Conservative Extensions in Guarded and Two-Variable Fragments*

Jean Christoph Jung[1], Carsten Lutz[1], Mauricio Martel[1], Thomas Schneider[1], and Frank Wolter[2]

1 University of Bremen, Bremen, Germany
{jeanjung,clu,martel,ts}@informatik.uni-bremen.de

2 University of Liverpool, Liverpool, UK
wolter@liverpool.ac.uk

Abstract

We investigate the decidability and computational complexity of (deductive) conservative extensions in fragments of first-order logic (FO), with a focus on the two-variable fragment FO^2 and the guarded fragment GF. We prove that conservative extensions are undecidable in any FO fragment that contains FO^2 or GF (even the three-variable fragment thereof), and that they are decidable and 2ExpTime-complete in the intersection GF^2 of FO^2 and GF.

1 Introduction

Conservative extensions are a fundamental notion in logic. In mathematical logic, they provide an important tool for relating logical theories, such as theories of arithmetic and theories that emerge in set theory [38, 34]. In computer science, they come up in diverse areas such as software specification [14], higher order theorem proving [18], and ontologies [27]. In these applications, it would be very useful to decide, given two sentences φ_1 and φ_2, whether $\varphi_1 \wedge \varphi_2$ is a conservative extension of φ_1. As expected, this problem is undecidable in first-order logic (FO). In contrast, it has been observed in recent years that conservative extensions are decidable in many modal and description logics [16, 29, 30, 7]. This observation is particularly interesting from the viewpoint of ontologies, where conservative extensions have many natural applications including modularity and reuse, refinement, versioning, and forgetting [11, 27].

Regarding decidability, conservative extensions thus seem to behave similarly to the classical satisfiability problem, which is also undecidable in FO while it is decidable for modal and description logics. In the case of satisfiability, the aim to understand the deeper reasons for this discrepancy and to push the limits of decidability to more expressive fragments of FO has sparked a long line of research that identified prominent decidable FO fragments such as the two-variable fragment FO^2 [37, 32], its extension with counting quantifiers C^2 [22], the guarded fragment GF [1], and the guarded negation fragment GNF [4], see also [6, 19, 36, 26] and references therein. These fragments have sometimes been used as a replacement for the modal and description logics that they generalize, and in particular the guarded fragment has been proposed as an ontology language [3]. Motivated by this situation, the aim of the current paper is to study the following two questions:

1. Are conservative extensions decidable in relevant fragments of FO such as FO^2, C^2, GF, and GNF?
2. What are the deeper reasons for decidability of conservative extensions in modal and description logics and how far can the limits of decidability be pushed?

* Funded by DFG grant LU 1417/2.

To be more precise, we concentrate on *deductive* conservative extensions, that is, $\varphi_1 \wedge \varphi_2$ is a conservative extension of φ_1 if for every sentence ψ formulated in the signature of φ_1, $\varphi_1 \wedge \varphi_2 \models \psi$ implies $\varphi_1 \models \psi$. There is also a *model-theoretic* notion of conservative extension which says that $\varphi_1 \wedge \varphi_2$ is a conservative extension of φ_1 if every model of φ_1 can be extended to a model of φ_2 by interpreting the additional symbols in φ_2. Model-theoretic conservative extensions imply deductive conservative extensions, but the converse fails unless one works with a very expressive logic such as second-order logic [27]. In fact, model-theoretic conservative extensions are undecidable even for some very inexpressive description logics that include neither negation nor disjunction [28]. Deductive conservative extensions, as studied in this paper, are closely related to other important notions in logic, such as uniform interpolation [33, 40, 5]. For example, in logics that enjoy Craig interpolation, a decision procedure for conservative extensions can also be used to decide whether a given sentence φ_2 is a uniform interpolant of a given sentence φ_1 regarding the symbols used in φ_2.

Instead of concentrating only on conservative extensions, we also consider two related reasoning problems that we call Σ-entailment and Σ-inseparability, where Σ denotes a signature. The definitions are as follows: a sentence φ_1 Σ-entails a sentence φ_2 if for every sentence ψ formulated in Σ, $\varphi_2 \models \psi$ implies $\varphi_1 \models \psi$. This can be viewed as a more relaxed notion of conservative extension where it is not required that one sentence actually extends the other as in the conjunction $\varphi_1 \wedge \varphi_2$ used in the definition of conservative extensions. Two sentences φ_1, φ_2 are Σ-inseparable if they Σ-entail each other. We generally prove lower bounds for conservative extensions and upper bounds for Σ-entailment, in this way obtaining the same decidability and complexity results for all three problems.

Our first main result is that conservative extensions are undecidable in FO^2 and (the three-variable fragment of) GF, and in fact in all fragments of FO that contain at least one of the two; note that the latter is not immediate because the separating sentence ψ in the definition of conservative extensions ranges over all sentences from the considered fragment, giving greater separating power when we move to a larger fragment. The proofs are by reductions from the halting problem for two-register machines and a tiling problem, respectively. We note that undecidability of conservative extensions also implies that there is no extension of the logic in question in which consequence is decidable and that has effective uniform interpolation (in the sense that uniform interpolants exist and are computable). We then show as our second main result that, in the two-variable guarded fragment GF^2, Σ-entailment is decidable in 2ExpTime. Regarding the satisfiability problem, GF^2 behaves fairly similarly to modal and description logics. It is thus suprising that deciding Σ-entailment (and conservative extensions) in GF^2 turns out to be much more challenging than in most modal and description logics. There, the main approach to proving decidability of Σ-entailment is to first establish a suitable model-theoretic characterization based on bisimulations which is then used as a foundation for a decision procedure based on tree automata [30, 7]. In GF^2, an analogous characterization in terms of appropriate guarded bisimulation fails. Instead, one has to demand the existence of *k-bounded* (guarded) bisimulations, *for all k*, and while tree automata can easily handle bisimulations, it is not clear how they can deal with such an infinite family of bounded bisimulations. We solve this problem by a very careful analysis of the situation and by providing another characterization that can be viewed as being 'half way' between a model-theoretic characterization and an automata-encoding of Σ-entailment.

We also observe that a 2ExpTime lower bound from [16] for conservative extensions in description logics can be adapted to GF^2, and consequently our upper bound is tight. It is known that GF^2 enjoys Craig interpolation and thus our results are also relevant to deciding uniform interpolants and to a stronger version of conservative extensions in which the separating sentence ψ can also use 'helper symbols' that occur neither in φ_1 nor in φ_2.

2 Preliminaries

We introduce the fragments of classical first-order logic (FO) that are relevant for this paper. We generally admit equality and disallow function symbols and constants. With *FO*^{2}, we denote the *two-variable fragment of FO*, obtained by fixing two variables x and y and disallowing the use of other variables [37, 32]. In FO^2 and fragments thereof, we generally admit only predicates of arity one and two, which is without loss of generality [20]. In the *guarded fragment of FO*, denoted *GF*, quantification is restricted to the pattern

$$\forall \mathbf{y}(\alpha(\mathbf{x},\mathbf{y}) \rightarrow \varphi(\mathbf{x},\mathbf{y})) \quad \exists \mathbf{y}(\alpha(\mathbf{x},\mathbf{y}) \wedge \varphi(\mathbf{x},\mathbf{y}))$$

where $\varphi(\mathbf{x},\mathbf{y})$ is a GF formula with free variables among $\mathbf{x},\mathbf{y}$ and $\alpha(\mathbf{x},\mathbf{y})$ is an atomic formula $R\mathbf{xy}$ or an equality $x = y$ that in either case contains all variables in $\mathbf{x},\mathbf{y}$ [1, 19]. The formula α is called the *guard* of the quantifier. The k-variable fragment of GF, defined in the expected way, is denoted GF^k. Apart from the logics introduced so far, in informal contexts we shall also mention several related description logics. Exact definitions are omitted, we refer the reader to [2].

A *signature* Σ is a finite set of predicates. We use GF(Σ) to denote the set of all GF-sentences that use only predicates from Σ (and possibly equality), and likewise for GF$^2(\Sigma)$ and other fragments. We use $\mathsf{sig}(\varphi)$ to denote the set of predicates that occur in the FO formula φ. Note that we consider equality to be a logical symbol, rather than a predicate, and it is thus never part of a signature. We write $\varphi_1 \models \varphi_2$ if φ_2 is a logical consequence of φ_1. The next definition introduces the central notions studied in this paper.

▶ **Definition 1.** Let F be a fragment of FO, φ_1, φ_2 F-sentences and Σ a signature. Then

1. φ_1 Σ*-entails* φ_2, written $\varphi_1 \models_\Sigma \varphi_2$, if for all $F(\Sigma)$-sentences ψ, $\varphi_2 \models \psi$ implies $\varphi_1 \models \psi$;
2. φ_1 and φ_2 are Σ*-inseparable* if φ_1 Σ-entails φ_2 and vice versa;
3. $\varphi_1 \wedge \varphi_2$ is a *conservative extension* of φ_1 if φ_1 $\mathsf{sig}(\varphi_1)$-entails $\varphi_1 \wedge \varphi_2$.

Note that Σ-entailment could equivalently be defined as follows when F is closed under negation: φ_1 Σ*-entails* φ_2 if for all $F(\Sigma)$-sentences ψ, satisfiability of $\varphi_1 \wedge \psi$ implies satisfiability of $\varphi_2 \wedge \psi$. If φ_1 does not Σ-entail φ_2, there is thus an $F(\Sigma)$-sentence ψ such that $\varphi_1 \wedge \psi$ is satisfiable while $\varphi_2 \wedge \psi$ is not. We refer to such ψ as a *witness sentence* for non-Σ-entailment.

▶ **Example 2.** (1) Σ-entailment is a weakening of logical consequence, that is, $\varphi_1 \models \varphi_2$ implies $\varphi_1 \models_\Sigma \varphi_2$ for any Σ. The converse is true when $\mathsf{sig}(\varphi_2) \subseteq \Sigma$.

(2) Consider the GF^2 sentences $\varphi_1 = \forall x \exists y Rxy$ and $\varphi_2 = \forall x(\exists y(Rxy \wedge Ay) \wedge \exists y(Rxy \wedge \neg Ay))$ and let $\Sigma = \{R\}$. Then $\psi = \forall xy(Rxy \rightarrow x = y)$ is a witness for $\varphi_1 \not\models_\Sigma \varphi_2$. If φ_1 is replaced by $\varphi_1' = \forall x \exists y(Rxy \wedge x \neq y)$ we obtain $\varphi_1' \models_\Sigma \varphi_2$ since GF^2 cannot count the number of R-successors.

It is important to note that different fragments F of FO give rise to different notions of Σ-entailment, Σ-inseparability and conservative extensions. For example, if φ_1 and φ_2 belong to GF^2, then they also belong to GF and to FO^2, but it might make a difference whether witness sentences range over all GF^2-sentences, over all GF-sentences, or over all FO^2-sentences. If we want to emphasize the fragment F in which witness sentences are formulated, we speak of $F(\Sigma)$-entailment instead of Σ-entailment and write $\varphi_1 \models_{F(\Sigma)} \varphi_2$, and likewise for $F(\Sigma)$-inseparability and F-conservative extensions.

► **Example 3.** Let φ_1', φ_2, and $\Sigma = \{R\}$ be from Example 2 (2). Then φ_1' $\mathrm{GF}^2(\Sigma)$-entails φ_2 but φ_1' does not $\mathrm{FO}(\Sigma)$-entail φ_2; a witness is given by $\forall xy_1y_2((Rxy_1 \wedge Rxy_2) \rightarrow y_1 = y_2)$.

Note that conservative extensions and Σ-inseparability reduce in polynomial time to Σ-entailment (with two calls to Σ-entailment required in the case of Σ-inseparability). Moreover, conservative extensions reduce in polynomial time to Σ-inseparability. We thus state our upper bounds in terms of Σ-entailment and lower bounds in terms of conservative extensions.

There is a natural variation of each of the three notions in Definition 1 obtained by allowing to use additional 'helper predicates' in witness sentences. For a fragment F of FO, F-sentences φ_1, φ_2, and a signature Σ, we say that φ_1 *strongly* Σ*-entails* φ_2 if φ_1 Σ'-entails φ_2 for any Σ' with $\Sigma' \cap \mathsf{sig}(\varphi_2) \subseteq \Sigma$. Strong Σ-inseparability and strong conservative extensions are defined accordingly. Strong Σ-entailment implies Σ-entailment, but the converse may fail.

► **Example 4.** $\mathrm{GF}(\Sigma)$-entailment does not imply strong $\mathrm{GF}(\Sigma)$-entailment. Let φ_1 state that the binary predicate R is irreflexive and symmetric and let φ_2 be the conjunction of φ_1 and $\forall x(Ax \rightarrow \forall y(Rxy \rightarrow \neg Ay)) \wedge \forall x(\neg Ax \rightarrow \forall y(Rxy \rightarrow Ay))$. Thus, an $\{R\}$-structure satisfying φ_1 can be extended to a model of φ_2 if it contains no R-cycles of odd length. Now observe that any satisfiable $\mathrm{GF}(\{R\})$ sentence is satisfiable in a forest $\{R\}$-structure (see Section 4 for a precise definition). Hence, if a $\mathrm{GF}(\{R\})$-sentence is satisfiable in an irreflexive and symmetric structure then it is satisfiable in a structure without odd cycles and so φ_1 $\mathrm{GF}(\{R\})$-entails φ_2. In contrast, for the fresh ternary predicate Q and $\psi = \exists x_1x_2x_3(Qx_1x_2x_3 \wedge Rx_1x_2 \wedge Rx_2x_3 \wedge Rx_3x_1)$ we have $\varphi_2 \models \neg\psi$ but $\varphi_1 \not\models \neg\psi$ and so ψ witnesses that φ_1 does not $\mathrm{GF}(\{R, Q\})$-entail φ_2.

The example above is inspired by proofs that GF does not enjoy Craig interpolation [24, 13]. This is not accidental, as we explain next. Recall that a fragment F of FO *has Craig interpolation* if for all F-sentences ψ_1, ψ_2 with $\psi_1 \models \psi_2$ there exists an F-sentence ψ (called an F*-interpolant for* ψ_1, ψ_2) such that $\psi_1 \models \psi \models \psi_2$ and $\mathsf{sig}(\psi) \subseteq \mathsf{sig}(\psi_1) \cap \mathsf{sig}(\psi_2)$. F *has uniform interpolation* if one can always choose an F-interpolant that does not depend on ψ_2, but only on ψ_1 and $\mathsf{sig}(\psi_1) \cap \mathsf{sig}(\psi_2)$. Thus, given ψ_1, ψ and Σ with $\psi_1 \models \psi$ and $\mathsf{sig}(\psi) \subseteq \Sigma$, then ψ is a *uniform* $F(\Sigma)$*-interpolant of* ψ_1 iff ψ strongly $F(\Sigma)$-entails ψ_1. Both Craig interpolation and uniform interpolation have been investigated extensively, for example for intuitionistic logic [33], modal logics [40, 12, 31], guarded fragments [13], and description logics [30]. The following observation summarizes the connection between (strong) Σ-entailment and interpolation.

► **Theorem 5.** *Let F be a fragment of FO that enjoys Craig interpolation. Then $F(\Sigma)$-entailment implies strong $F(\Sigma)$-entailment. In particular, if $\varphi_2 \models \varphi_1$ and $\mathsf{sig}(\varphi_1) \subseteq \Sigma$, then φ_1 is a uniform $F(\Sigma)$-interpolant of φ_2 iff φ_1 $F(\Sigma)$-entails φ_2.*

Proof. Assume φ_1 does not strongly $F(\Sigma)$-entail φ_2. Then there is an F-sentence ψ with $\mathsf{sig}(\psi) \cap \mathsf{sig}(\varphi_2) \subseteq \Sigma$ such that $\varphi_2 \models \psi$ and $\varphi_1 \wedge \neg\psi$ is satisfiable. Let χ be an interpolant for φ_2 and ψ in F. Then $\neg\chi$ witnesses that φ_1 does not $F(\Sigma)$-entail φ_2. ❏

The *uniform interpolant recognition problem for F* is the problem to decide whether a sentence ψ is a uniform $F(\Sigma)$-interpolant of a sentence ψ'. It follows from Theorem 5 that in any fragment F of FO that enjoys Craig interpolation, this problem reduces in polynomial time to Σ-inseparability in F and that, conversely, conservative extension in F reduces in polynomial time to the uniform interpolant recognition problem in F. Neither GF nor FO^2 nor description logics with role inclusions enjoy Craig interpolation [24, 10, 27], but GF^2

does [24]. Thus, our decidability and complexity results for Σ-entailment in GF^2 also apply to strong Σ-entailment and the uniform interpolant recognition problem.

3 Undecidability

We prove that conservative extensions are undecidable in GF^3 and in FO^2, and consequently so are Σ-entailment and Σ-inseparability (as well as strong Σ-entailment and the uniform interpolant recognition problem). These results hold already without equality and in fact apply to all fragments of FO that contain at least one of GF^3 and FO^2 such as the guarded negation fragment [4] and the two-variable fragment with counting quantifiers [22].

We start with the case of GF^3, using a reduction from the halting problem of two-register machines. A (deterministic) *two-register machine (2RM)* is a pair $M = (Q, P)$ with $Q = q_0, \dots, q_\ell$ a set of *states* and $P = I_0, \dots, I_{\ell-1}$ a sequence of *instructions*. By definition, q_0 is the *initial state*, and q_ℓ the *halting state*. For all $i < \ell$,

- either $I_i = +(p, q_j)$ is an *incrementation instruction* with $p \in \{0, 1\}$ a register and q_j the subsequent state;
- or $I_i = -(p, q_j, q_k)$ is a *decrementation instruction* with $p \in \{0, 1\}$ a register, q_j the subsequent state if register p contains 0, and q_k the subsequent state otherwise.

A *configuration* of M is a triple (q, m, n), with q the current state and $m, n \in \mathbb{N}$ the register contents. We write $(q_i, n_1, n_2) \Rightarrow_M (q_j, m_1, m_2)$ if one of the following holds:

- $I_i = +(p, q_j)$, $m_p = n_p + 1$, and $m_{1-p} = n_{1-p}$;
- $I_i = -(p, q_j, q_k)$, $n_p = m_p = 0$, and $m_{1-p} = n_{1-p}$;
- $I_i = -(p, q_k, q_j)$, $n_p > 0$, $m_p = n_p - 1$, and $m_{1-p} = n_{1-p}$.

The *computation* of M on input $(n, m) \in \mathbb{N}^2$ is the unique longest configuration sequence $(p_0, n_0, m_0) \Rightarrow_M (p_1, n_1, m_1) \Rightarrow_M \cdots$ such that $p_0 = q_0$, $n_0 = n$, and $m_0 = m$. The halting problem for 2RMs is to decide, given a 2RM M, whether its computation on input $(0, 0)$ is finite (which implies that its last state is q_ℓ).

We show how to convert a given 2RM M into GF^3-sentences φ_1 and φ_2 such that M halts on input $(0, 0)$ iff $\varphi_1 \wedge \varphi_2$ is not a conservative extension of φ_1. Let $M = (Q, P)$ with $Q = q_0, \dots, q_\ell$ and $P = I_0, \dots, I_{\ell-1}$. We assume w.l.o.g. that $\ell \geq 1$ and that if $I_i = -(p, q_j, q_k)$, then $q_j \neq q_k$. In φ_1, we use the following set Σ of predicates:

- a binary predicate N connecting a configuration to its successor configuration;
- binary predicates R_1 and R_2 that represent the register contents via the length of paths;
- unary predicates $q_0, \dots, q_\ell$ representing the states of M;
- a unary predicate S denoting points where a computation starts.

We define φ_1 to be the conjunction of several GF^2-sentences. First, we say that there is a point where the computation starts:[1]

$$\exists x Sx \wedge \forall x(Sx \rightarrow (q_0 x \wedge \neg\exists y\, R_0 xy \wedge \neg\exists y\, R_1 xy))$$

And second, we add that whenever M is not in the final state, there is a next configuration. For $0 \leq i < \ell$:

$$\begin{aligned}
\forall x(q_i x \rightarrow \exists y(Nxy \wedge q_j y)) &\quad \text{if } I_i = +(p, q_j) \\
\forall x((q_i x \wedge \neg\exists y R_p xy) \rightarrow \exists y(Nxy \wedge q_j y)) &\quad \text{if } I_i = -(p, q_j, q_k) \\
\forall x((q_i x \wedge \exists y R_p xy) \rightarrow \exists y(Nxy \wedge q_k y)) &\quad \text{if } I_i = -(p, q_j, q_k)
\end{aligned}$$

---

[1] The formulas that are not syntactically guarded can easily be rewritten into such formulas.

The second sentence φ_2 is constructed so as to express that either M does not halt or the representation of the computation of M contains a defect, using the following additional predicates:

- a unary predicate P used to represent that M does not halt;
- binary predicates $D_p^+, D_p^-, D_p^=$ used to describe defects in incrementing, decrementing, and keeping register $p \in \{0, 1\}$;
- ternary predicates $H_1^+, H_2^+, H_1^-, H_2^-, H_1^=, H_2^=$ used as guards for existential quantifiers.

In fact, φ_2 is the disjunction of two sentences. The first sentence says that the computation does not terminate:

$$\exists x\,(Sx \wedge Px) \wedge \forall x\,(Px \rightarrow \exists y\,(Nxy \wedge Py))$$

while the second says that registers are not updated properly:

$$\begin{aligned}
\exists x \exists y \big(Nxy \wedge (& \bigvee_{I_i=+(p,q_j)} (q_i x \wedge q_j y \wedge (D_p^+ xy \vee D_{1-p}^= xy)) \\
& \vee \bigvee_{I_i=-(p,q_j,q_k)} (q_i x \wedge q_k y \wedge (D_p^- xy \vee D_{1-p}^= xy)) \\
& \vee \bigvee_{I_i=-(p,q_j,q_k)} (q_i x \wedge q_j y \wedge (D_p^= xy \vee D_{1-p}^= xy)))\big) \\
\wedge \forall x \forall y\,(D_p^+ xy \rightarrow (& \neg \exists z\, R_p yz \vee (\neg \exists z\, R_p xz \wedge \exists z\,(R_p yz \wedge \exists x R_p zx)) \\
& \vee \exists z (H_1^+ xyz \wedge R_p xz \wedge \exists x (H_2^+ xzy \wedge R_p yx \wedge D_p^+ zx)).
\end{aligned}$$

In this second sentence, additional conjuncts that implement the desired behaviour of $D_p^=$ and D_p^- are also needed; they are constructed analogously to the last three lines above (but using guards H_j^- and $H_j^=$), details are omitted. The following is proved in the appendix of this paper.

▶ **Lemma 6.**

1. *If M halts, then $\varphi_1 \wedge \varphi_2$ is not a GF^2-conservative extension of φ_1.*
2. *If there exists a Σ-structure that satisfies φ_1 and cannot be extended to a model of φ_2 (by interpreting the predicates in $\mathsf{sig}(\varphi_2) \setminus \mathsf{sig}(\varphi_1)$), then M halts.*

In the proof of Point 1, the sentence that witnesses non-conservativity describes a halting computation of M, up to global $\mathrm{GF}^2(\Sigma)$-bisimulations. This can be done using only two variables. The following result is an immediate consequence of Lemma 6.

▶ **Theorem 7.** *In any fragment of FO that extends the three-variable guarded fragment GF^3, the following problems are undecidable: conservative extensions, Σ-inseparability, Σ-entailment, and strong Σ-entailment.*

Since Point 1 of Lemma 6 ensures GF^2-witnesses, Theorem 7 can actually be strengthened to say that GF^2-conservative extensions of GF^3-sentences are undecidable.

Our result for FO^2 is proved by a reduction of a tiling problem that asks for the tiling of a rectangle (of any size) such that the borders are tiled with certain distinguished tiles. Because of space limitations, we defer details to the appendix and state only the obtained result.

▶ **Theorem 8.** *In any fragment of FO that extends FO^2, the following problems are undecidable: conservative extensions, Σ-inseparability, Σ-entailment, and strong Σ-entailment.*

It is interesting to note that the proof of Theorem 8 also shows that FO^2-conservative extensions of $\mathcal{ALC}$-TBoxes are undecidable while it follows from our results below that GF^2-conservative extensions of $\mathcal{ALC}$-TBoxes are decidable.

4 Characterizations

The undecidability results established in the previous section show that neither the restriction to two variables nor guardedness alone are sufficient for decidability of conservative extensions and related problems. In the remainder of the paper, we show that adopting both restrictions simultaneously results in decidability of Σ-entailment (and thus also of conservative extensions and of inseparability). We proceed by first establishing a suitable model-theoretic characterization and then use it as the foundation for a decision procedure based on tree automata. We in fact establish two versions of the characterization, the second one building on the first one.

We start with some preliminaries. An *atomic 1-type for* Σ is a maximal satisfiable set τ of atomic $\mathrm{GF}^2(\Sigma)$-formulas and their negations that use the variable x, only. We use $\mathrm{at}^{\Sigma}_{\mathfrak{A}}(a)$ to denote the atomic 1-type for Σ realized by the element a in the structure $\mathfrak{A}$. An *atomic 2-type for* Σ is a maximal satisfiable set τ of atomic $\mathrm{GF}^2(\Sigma)$-formulas and their negations that use the variables x and y, only, and contains $\neg(x = y)$. We say that τ is *guarded* if it contains an atom of the form Rxy or Ryx, R a predicate symbol. We use $\mathrm{at}^{\Sigma}_{\mathfrak{A}}(a, b)$ to denote the atomic 2-type for Σ realized by the elements a, b in the structure $\mathfrak{A}$. A relation $\sim\, \subseteq A \times B$ is a *$GF^2(\Sigma)$-bisimulation between $\mathfrak{A}$ and $\mathfrak{B}$* if the following conditions hold whenever $a \sim b$:

1. $\mathrm{at}^{\Sigma}_{\mathfrak{A}}(a) = \mathrm{at}^{\Sigma}_{\mathfrak{B}}(b)$;
2. for every $a' \neq a$ such that $\mathrm{at}^{\Sigma}_{\mathfrak{A}}(a, a')$ is guarded, there is a $b' \neq b$ such that $\mathrm{at}^{\Sigma}_{\mathfrak{A}}(a, a') = \mathrm{at}^{\Sigma}_{\mathfrak{B}}(b, b')$ and $a' \sim b'$ (forth condition);
3. for every $b' \neq b$ such that $\mathrm{at}^{\Sigma}_{\mathfrak{B}}(b, b')$ is guarded, there is an $a' \neq a$ such that $\mathrm{at}^{\Sigma}_{\mathfrak{A}}(a, a') = \mathrm{at}^{\Sigma}_{\mathfrak{B}}(b, b')$ and $a' \sim b'$ (back condition).

We write $(\mathfrak{A}, a) \sim_{\Sigma} (\mathfrak{B}, b)$ and say that $(\mathfrak{A}, a)$ and $(\mathfrak{B}, b)$ are *$GF^2(\Sigma)$-bisimilar* if there is a $\mathrm{GF}^2(\Sigma)$-bisimulation $\sim$ between $\mathfrak{A}$ and $\mathfrak{B}$ with $a \sim b$. If the domain and range of $\sim$ coincide with A and B, respectively, then $\sim$ is called a *global $GF^2(\Sigma)$-bisimulation*.

We next introduce a bounded version of bisimulations. For $k \geq 0$, we write $(\mathfrak{A}, a) \sim^k_{\Sigma} (\mathfrak{B}, b)$ and say that $(\mathfrak{A}, a)$ and $(\mathfrak{B}, b)$ are *k-$GF^2(\Sigma)$-bisimilar* if there is a $\sim\, \subseteq A \times B$ such that the first condition for bisimulations holds and the back and forth conditions can be iterated up to k times starting from a and b; a formal definition is in the appendix. It is straightforward to show the following link between k-GF^2-bisimilarity and GF^2-sentences of guarded quantifier depth k (defined in the obvious way).

▶ **Lemma 9.** *Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures, Σ a signature, and $k \geq 0$. Then the following conditions are equivalent:*

1. *for all $a \in A$ there exists $b \in B$ with $(\mathfrak{A}, a) \sim^k_{\Sigma} (\mathfrak{B}, b)$ and vice versa;*
2. *$\mathfrak{A}$ and $\mathfrak{B}$ satisfy the same $GF^2(\Sigma)$-sentences of guarded quantifier depth at most k.*

The corresponding lemma for $\mathrm{GF}^2(\Sigma)$-sentences of unbounded guarded quantifier depth and $\mathrm{GF}^2(\Sigma)$-bisimulations holds if $\mathfrak{A}$ and $\mathfrak{B}$ satisfy certain saturation conditions (for example, if $\mathfrak{A}$ and $\mathfrak{B}$ are ω-saturated). It can then be proved that an FO-sentence φ is equivalent to a GF^2 sentence iff its models are preserved under global $\mathrm{GF}^2(\mathsf{sig}(\varphi))$-bisimulations [21, 17]. In modal and description logic, global Σ-bisimulations can often be used to characterize Σ-entailment in the following natural way [30]: φ_1 Σ-entails φ_2 iff every for every (tree) model $\mathfrak{A}$ of φ_1, there exists a (tree) model $\mathfrak{B}$ of φ_2 that is globally Σ-bisimilar to $\mathfrak{A}$. Such a characterization enables decision procedures based on tree automata, but does not hold for GF^2.

▶ **Example 10.** Let $\varphi_1 = \forall x \exists y Rxy$ and let $\varphi_2 = \varphi_1 \wedge \exists x Bx \wedge \forall x(Bx \to \exists y(Ryx \wedge By))$. Let $\mathfrak{A}$ be the model of φ_1 that consists of an infinite R-path with an initial element. Then there is no model of φ_2 that is globally $\mathrm{GF}^2(\{R\})$-bisimilar to $\mathfrak{A}$ since any such model has to contain an infinite R-path with no initial element. Yet, φ_2 is a conservative extension of φ_1 which can be proved using Theorem 11 below.

We give our first characterization theorem that uses unbounded bisimulations in one direction and bounded bisimulations in the other.

▶ **Theorem 11.** *Let φ_1, φ_2 be GF_2-sentences and Σ a signature. Then $\varphi_1 \models_\Sigma \varphi_2$ iff for every model $\mathfrak{A}$ of φ_1 of finite outdegree, there is a model $\mathfrak{B}$ of φ_2 such that*
1. *for every $a \in A$ there is a $b \in B$ such that $(\mathfrak{A}, a) \sim_\Sigma (\mathfrak{B}, b)$*
2. *for every $b \in B$ and every $k \geq 0$, there is an $a \in A$ such that $(\mathfrak{A}, a) \sim^k_\Sigma (\mathfrak{B}, b)$.*

The direction ($\Leftarrow$) follows from Lemma 9 and ($\Rightarrow$) can be proved using compactness and ω-saturated structures. Because of the use of k-bounded bisimulations (for unbounded k), it is not clear how to use Theorem 11 to find a decision procedure based on tree automata. In the following, we formulate a more 'operational' but also more technical characterization that no longer mentions bounded bisimulations. It additionally refers to forest models $\mathfrak{A}$ of φ_1 (of finite outdegree) instead of unrestricted models, but we remark that Theorem 11 also remains true under this modification.

A structure $\mathfrak{A}$ is a *forest* if its Gaifman graph is a forest. Thus, a forest admits cycles of length one and two, but not of any higher length. A (Σ-)*tree* in a forest structure $\mathfrak{A}$ is a maximal (Σ)-connected substructure of $\mathfrak{A}$. When working with forest structures $\mathfrak{A}$, we will typically view them as directed forests rather than as undirected ones. This can be done by choosing a root for each tree in the Gaifman graph of $\mathfrak{A}$, thus giving rise to notions such as successor, descendant, etc. Which node is chosen as the root will always be irrelevant. Note that the direction of binary relations does not need to reflect the successor relation. When speaking of a *path* in a forest structure $\mathfrak{A}$, we mean a path in the directed sense; when speaking of a *subtree*, we mean a tree that is obtained by choosing a root a and restricting the structure to a and its descendants. We say that $\mathfrak{A}$ is *regular* if it has only finitely many subtrees, up to isomorphism.

To see how we can get rid of bounded bisimulations, reconsider Theorem 11. The characterization is still correct if we pull out the quantification over k in Point 2 so that the theorem reads '...iff for every model $\mathfrak{A}$ of φ_1 of finite outdegree and every $k \geq 0$, there is...'. In fact, this modified version of Theorem 11 is even closer to the definition of Σ-entailment. It also suggests that we add a marking $A_\perp \subseteq A$ of elements in $\mathfrak{A}$, representing 'break-off points' for bisimulations, and then replace k-bisimulations with bisimulations that stop whenever they have encountered the *second* marked element on the same path—in this way, the distance between marked elements (roughly) corresponds to the bound k in k-bisimulations. However, we would need a marking $A_\perp$, for any $k \geq 0$, such that there are infinitely many markers on any infinite path and the distance between any two markers in a tree is at least k. It is easy to see that such a marking may not exist, for example when $k = 3$ and $\mathfrak{A}$ is the infinite full binary tree. We solve this problem as follows. First, we only demand that the distance between any two markers *on the same path* is at least k. And second, we use the markers only when following bisimulations upwards in a tree while downwards, we use unbounded bisimulations. This does not compromise correctness of the characterization.

We next introduce a version of bisimulations that implement the ideas just explained. Let $\mathfrak{A}$ and $\mathfrak{B}$ be forest models, Σ a signature, and $A_\perp \subseteq A$. Two relations $\sim^{A_\perp,0}_\Sigma, \sim^{A_\perp,1}_\Sigma \subseteq A \times B$ form an *$A_\perp$-delimited $GF^2(\Sigma)$-bisimulation* between $\mathfrak{A}$ and $\mathfrak{B}$ if the following conditions are satisfied:
1. if $(\mathfrak{A}, a) \sim^{A_\perp,0}_\Sigma (\mathfrak{B}, b)$, then $\mathrm{at}^\Sigma_\mathfrak{A}(a) = \mathrm{at}^\Sigma_\mathfrak{B}(b)$ and

   a. for every $a' \neq a$ with $\mathrm{at}^{\Sigma}_{\mathfrak{A}}(a, a')$ guarded, there is a $b' \neq b$ such that $(\mathfrak{A}, a') \sim^{A_\perp,i}_{\Sigma} (\mathfrak{B}, b')$ where $i = 1$ if a' is the predecessor of a and $a' \in A_\perp$, and $i = 0$ otherwise;
   b. for every $b' \neq b$ with $\mathrm{at}^{\Sigma}_{\mathfrak{B}}(b, b')$ guarded, there is an $a' \neq a$ such that $(\mathfrak{A}, a') \sim^{A_\perp,i}_{\Sigma} (\mathfrak{B}, b')$ where $i = 1$ if a' is the predecessor of a and $a' \in A_\perp$, and $i = 0$ otherwise;

2. if $(\mathfrak{A}, a) \sim^{A_\perp,1}_{\Sigma} (\mathfrak{B}, b)$ and the predecessor of a in $\mathfrak{A}$ is not in $A_\perp$, then $\mathrm{at}^{\Sigma}_{\mathfrak{A}}(a) = \mathrm{at}^{\Sigma}_{\mathfrak{B}}(b)$ and
   a. for every $a' \neq a$ with $\mathrm{at}^{\Sigma}_{\mathfrak{A}}(a, a')$ guarded, there is a $b' \neq b$ such that $(\mathfrak{A}, a') \sim^{A_\perp,i}_{\Sigma} (\mathfrak{B}, b')$ where $i = 0$ if a is the predecessor of a' and $a \in A_\perp$, and $i = 1$ otherwise;
   b. for every $b' \neq b$ with $\mathrm{at}^{\Sigma}_{\mathfrak{B}}(b, b')$ guarded, there is an $a' \neq a$ such that $(\mathfrak{A}, a') \sim^{A_\perp,i}_{\Sigma} (\mathfrak{B}, b')$ where $i = 0$ if a is the predecessor of a' and $a \in A_\perp$, and $i = 1$ otherwise.

Then $(\mathfrak{A}, a)$ and $(\mathfrak{B}, b)$ are *$A_\perp$-delimited GF$^2(\Sigma)$-bisimilar*, in symbols $(\mathfrak{A}, a) \sim^{A_\perp}_{\Sigma} (\mathfrak{B}, b)$, if there exists an $A_\perp$-delimited GF$^2(\Sigma)$-bisimulation $\sim^{A_\perp,0}_{\Sigma}, \sim^{A_\perp,1}_{\Sigma}$ between $\mathfrak{A}$ and $\mathfrak{B}$ such that $(\mathfrak{A}, a) \sim^{A_\perp,0}_{\Sigma} (\mathfrak{B}, b)$.

Let φ be a GF^2-sentence. We use $\mathsf{cl}(\varphi)$ to denote the set of all subformulas of φ closed under single negation and renaming of free variables (using only the available variables x and y). A 1-*type for* φ is a subset $t \subseteq \mathsf{cl}(\varphi)$ that contains only formulas of the form $\psi(x)$ and such that $\varphi \wedge \exists x \bigwedge t(x)$ is satisfiable. For a model $\mathfrak{A}$ of φ and $a \in A$, we use $\mathsf{tp}_{\mathfrak{A}}(a)$ to denote the 1-type $\{\psi(x) \in \mathsf{cl}(\varphi) \mid \mathfrak{A} \models \psi(a)\}$, assuming that φ is understood from the context. We say that the 1-type t is *realized* in $\mathfrak{A}$ if there is an $a \in A$ with $\mathsf{tp}_{\mathfrak{A}}(a) = t$. We are now ready to formulate our final characterizations.

▶ **Theorem 12.** *Let φ_1, φ_2 be GF^2-sentences and Σ a signature. Then $\varphi_1 \models_\Sigma \varphi_2$ iff for every regular forest model $\mathfrak{A}$ of φ_1 that has finite outdegree and for every set $A_\perp \subseteq A$ with $A_\perp \cap \rho$ infinite for any infinite Σ-path ρ in $\mathfrak{A}$, there is a model $\mathfrak{B}$ of φ_2 such that*
1. *for every $a \in A$, there is a $b \in B$ such that $(\mathfrak{A}, a) \sim_\Sigma (\mathfrak{B}, b)$;*
2. *for every 1-type t for φ_2 that is realized in $\mathfrak{B}$, there are $a \in A$ and $b \in B$ such that $\mathsf{tp}_{\mathfrak{B}}(b) = t$ and $(\mathfrak{A}, a) \sim^{A_\perp}_{\Sigma} (\mathfrak{B}, b)$.*

Regularity and finite outdegree are used in the proof of Theorem 12 given in the appendix, but it follows from the automata constructions below that the theorem is still correct when these qualifications are dropped.

5 Decidability and Complexity

We show that Σ-entailment in GF^2 is decidable and 2ExpTime-complete, and thus so are conservative extensions and Σ-inseparability. The upper bound is based on Theorem 12 and uses alternating parity automata on infinite trees. Since Theorem 12 does not provide us with an obvious upper bound on the outdegree of the involved tree models, we use alternating tree automata which can deal with trees of any finite outdegree, similar to the ones introduced by Wilke [41], but with the capability to move both downwards and upwards in the tree.

A *tree* is a non-empty (and potentially infinite) set of words $T \subseteq (\mathbb{N} \setminus 0)^*$ closed under prefixes. We generally assume that trees are finitely branching, that is, for every $w \in T$, the set $\{i \mid w \cdot i \in T\}$ is finite. For any $w \in (\mathbb{N} \setminus 0)^*$, as a convention we set $w \cdot 0 := w$. If $w = n_0 n_1 \cdots n_k$, we additionally set $w \cdot -1 := n_0 \cdots n_{k-1}$. For an alphabet Θ, a Θ-*labeled tree* is a pair (T, L) with T a tree and $L : T \to \Theta$ a node labeling function.

A *two-way alternating tree automata (2ATA)* is a tuple $\mathcal{A} = (Q, \Theta, q_0, \delta, \Omega)$ where Q is a finite set of *states*, Θ is the *input alphabet*, $q_0 \in Q$ is the *initial state*, δ is a *transition function* as specified below, and $\Omega : Q \to \mathbb{N}$ is a *priority function*, which assigns a priority to each state. The transition function maps a state q and some input letter $\theta \in \Theta$ to a *transition* *condition* $\delta(q, \theta)$ which is a positive Boolean formula over the truth constants true and false and transitions of the form q, $\langle - \rangle q$, $[-]q$, $\Diamond q$, $\Box q$ where $q \in Q$. The automaton runs on Θ-labeled trees. Informally, the transition q expresses that a copy of the automaton is sent to the current node in state q, $\langle - \rangle q$ means that a copy is sent in state q to the predecessor node, which is then required to exist, $[-]q$ means the same except that the predecessor node is not required to exist, $\Diamond q$ means that a copy is sent in state q to some successor, and $\Box q$ that a copy is sent in state q to all successors. The semantics is defined in terms of runs in the usual way, we refer to the appendix for details. We use $L(\mathcal{A})$ to denote the set of all Θ-labeled trees accepted by $\mathcal{A}$. It is standard to verify that 2ATAs are closed under complementation and intersection. We show in the appendix that the emptiness problem for 2ATAs can be solved in time exponential in the number of states.

We aim to show that given two GF^2-sentences φ_1 and φ_2 and a signature Σ, one can construct a 2ATA $\mathcal{A}$ such that $L(\mathcal{A}) = \emptyset$ iff $\varphi_1 \models_{\mathrm{GF}^2(\Sigma)} \varphi_2$. The number of states of the 2ATA $\mathcal{A}$ is polynomial in the size of φ_1 and exponential in the size of φ_2, which yields the desired 2ExpTime upper bounds.

Let φ_1, φ_2, and Σ be given. Since the logics we are concerned with have Craig interpolation, we can assume w.l.o.g. that $\Sigma \subseteq \mathsf{sig}(\varphi_1)$. With Θ, we denote the set of all pairs (τ, M) where τ is an atomic 2-type for $\mathsf{sig}(\varphi_1)$ and $M \in \{0, 1\}$. For $p = (\tau, M) \in \Theta$, we use p^1 to denote τ and p^2 to denote M. A Θ-labeled tree (T, L) represents a forest structure $\mathfrak{A}_{(T,L)}$ with universe $A_{(T,L)} = T$ and where $w \in A^{\mathfrak{A}_{(T,L)}}$ if $A(y) \in L(w)$ and $(w, w') \in R^{\mathfrak{A}_{(T,L)}}$ if one of the following conditions is satisfied: (1) $w = w'$ and $Ryy \in L(w)^1$; (2) w' is a successor of w and $Rxy \in L(w')^1$; (3) w is a successor of w' and $Ryx \in L(w)^1$. Thus, the atoms in a node label that involve only the variable y describe the current node, the atoms that involve both variables x and y describe the connection between the predecessor and the current node, and the atoms that involve only the variable x are ignored. The M-components of node labels are used to represent a set of markers $A_\perp = \{w \in A_{(T,L)} \mid L(w)^2 = 1\}$. It is easy to see that, conversely, for every tree structure $\mathfrak{A}$ over Σ, there is a Θ-labeled tree (T, L) such that $\mathfrak{A}_{(T,L)} = \mathfrak{A}$.

To obtain the desired 2ATA $\mathcal{A}$, we construct three 2ATAs $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ and then define $\mathcal{A}$ so that it accepts $L(\mathcal{A}_1) \cap \overline{L(\mathcal{A}_2)} \cap L(\mathcal{A}_3)$. The 2ATA $\mathcal{A}_3$ only makes sure that the set $A_\perp \subseteq A_{(T,L)}$ is such that for any infinite Σ-path ρ, $A_\perp \cap \rho$ is infinite (as required by Theorem 12), we omit details. We construct $\mathcal{A}_1$ so that it accepts a Θ-labeled tree (T, L) iff $\mathfrak{A}_{(T,L)}$ is a model of φ_1. The details of the construction, which is fairly standard, can be found in the appendix. The number of states of $\mathcal{A}_1$ is polynomial in the size of φ_1 and independent of φ_2. The most interesting automaton is $\mathcal{A}_2$.

▶ **Lemma 13.** *There is a 2ATA $\mathcal{A}_2$ that accepts a Θ-labeled tree (T, L) iff there is a model $\mathfrak{B}$ of φ_2 s.t. Conditions 1 and 2 from Theorem 12 are satisfied when $\mathfrak{A}$ is replaced with $\mathfrak{A}_{(T,L)}$.*

The general idea of the construction of $\mathcal{A}_2$ is to check the existence of the desired model $\mathfrak{B}$ of φ_2 by verifying that there is a set of 1-types for φ_2 from which $\mathfrak{B}$ can be assembled, represented via the states that occur in a successful run. Before we can give details, we introduce some preliminaries.

A 0-*type* s *for* φ_2 is a maximal set of sentences $\psi() \in \mathsf{cl}(\varphi_2)$ such that $\varphi_2 \wedge s$ is satisfiable. A 2-*type* λ *for* φ_2 is a maximal set of formulas $\psi(x, y) \in \mathsf{cl}(\varphi_2)$ that contains $\neg(x = y)$ and such that $\varphi_2 \wedge \exists xy\, \lambda(x, y)$ is satisfiable. If a 2-type λ contains the atom Rxy or Ryx for at least one binary predicate R, then it is *guarded*. If additionally $R \in \Sigma$, then it is Σ-*guarded*. Note that each 1-type contains a (unique) 0-type and each 2-type contains two (unique) 1-types. Formally, we use λ_x to denote the 1-type obtained by restricting the 2-type λ to the formulas that do not use the variable y, and likewise for λ_y and the variable x. We use TP_n

to denote the set of n-types for φ_2, $n \in \{0,1,2\}$. For $t \in \mathsf{TP}_1$ and a $\lambda \in \mathsf{TP}_2$, we say that λ is *compatible with* t and write $t \approx \lambda$ if the sentence $\varphi_2 \wedge \exists xy(t(x) \wedge \lambda(x,y))$ is satisfiable; for $t \in \mathsf{TP}_1$ and $T \subseteq \mathsf{TP}_2$ a set of guarded 2-types, we say that T is a *neighborhood for* t and write $t \approx T$ if the sentence

$$\varphi_2 \wedge \exists x\big(t(x) \wedge \bigwedge_{\lambda \in T} \exists y\, \lambda(x,y) \wedge \forall y \bigvee_{R \in \mathsf{sig}(\varphi_2)} ((Rxy \vee Ryx) \rightarrow \bigvee_{\lambda \in T} \lambda(x,y)))$$

is satisfiable. Note that each of the mentioned sentences is formulated in GF^2 and at most single exponential in size (in the size of φ_1 and φ_2), thus satisfiability can be decided in 2ExpTime.

To build the automaton $\mathcal{A}_2$ from Lemma 13, set $\mathcal{A}_2 = (Q_2, \Theta, q_0, \delta_2, \Omega_2)$ where Q_2 is

$$\begin{array}{l}\{q_0, q_\perp\} \cup \mathsf{TP}_0 \cup \{t, t^?, t_\uparrow, t_\downarrow, t_\&, t^i, t^i_\uparrow, t^i_\downarrow \mid t \in \mathsf{TP}_1,\ i \in \{0,1\}\} \cup \\ \{\lambda, \lambda_\uparrow, \lambda^i, \lambda^i_\uparrow \mid \lambda \in \mathsf{TP}_2,\ i \in \{0,1\}\},\end{array}$$

Ω_2 assigns two to all states except for those of the form $t^?$, to which it assigns one.

The automaton begins by choosing the 0-type s realized in the forest model $\mathfrak{B}$ of φ_2 whose existence it aims to verify. For every $\exists x \varphi(x) \in s$, it then chooses a 1-type t in which $\varphi(x)$ is realized in $\mathfrak{B}$ and sends off a copy of itself to find a node where t is realized. To satisfy Condition 1 of Theorem 12, at each node it further chooses a 1-type that is compatible with s, to be realized at that node. This is implemented by the following transitions:

$$\begin{array}{rcl}\delta_2(q_0,\sigma) & = & \displaystyle\bigvee_{s \in \mathsf{TP}_0} \Big(s \wedge \bigwedge_{\exists x\, \varphi(x) \in s} \ \bigvee_{\substack{t \in \mathsf{TP}_1 \mid \\ s \cup \{\varphi(x)\} \subseteq t}} t^?\Big) \\ \delta_2(s,\sigma) & = & \Box s \wedge \displaystyle\bigvee_{t \in \mathsf{TP}_1, s \subseteq t} t \\ \delta_2(t^?,\sigma) & = & \langle -1 \rangle t^? \vee \Diamond t^? \vee t^0\end{array}$$

where s ranges over TP_0. When a state of the form t is assigned to a node w, this is an obligation to prove that there is a $\mathrm{GF}^2(\Sigma)$-bisimulation between the element w in $\mathfrak{A}_{(T,L)}$ and an element b of type t in $\mathfrak{B}$. A state of the form t^0 represents the obligation to verify that there is an $A_\perp$-delimited $\mathrm{GF}^2(\Sigma)$-bisimulation between w and an element of type t in $\mathfrak{B}$. We first verify that the former obligations are satisfied. This requires to follow all successors of w and to guess types of successors of b to be mapped there, satisfying the back condition of bisimulations. We also need to guess successors of b in $\mathfrak{B}$ (represented as a neighborhood for t) to satisfy the existential demands of t and then select successors of a to which they are mapped, satisfying the "back" condition of bisimulations. Whenever we decide to realize a 1-type t in $\mathfrak{B}$ that does not participate in the bisimulation currently being verified, we also send another copy of the automaton in state $t^?$ to guess an $a \in A_{(T,L)}$ that we can use to satisfy Condition 2 from Theorem 12:

$$\delta_2(t,(\tau,M)) \;=\; t_\uparrow \wedge \Box t_\downarrow \wedge \bigvee_{T \mid t \approx T} \bigwedge_{\lambda \in T} (\Diamond\lambda \vee \lambda_\uparrow) \quad \text{if } \tau_y =_\Sigma t$$

$$\begin{array}{rcll}
\delta_2(t,(\tau,M)) & = & \mathsf{false} & \text{if } \tau_y \neq_\Sigma t \\
\delta_2(t_\downarrow,(\tau,M)) & = & \mathsf{true} & \text{if } \tau \text{ is not } \Sigma\text{-guarded} \\
\delta_2(t_\downarrow,(\tau,M)) & = & \displaystyle\bigvee_{\lambda \mid t \approx \lambda \wedge \tau =_\Sigma \lambda} \lambda_y & \text{if } \tau \text{ is } \Sigma\text{-guarded} \\
\delta_2(t_\uparrow,(\tau,M)) & = & \mathsf{true} & \text{if } \tau \text{ is not } \Sigma\text{-guarded} \\
\delta_2(t_\uparrow,(\tau,M)) & = & \displaystyle\bigvee_{\lambda \mid t \approx \lambda \wedge \tau =_\Sigma \lambda^-} [-1]\lambda_y & \text{if } \tau \text{ is } \Sigma\text{-guarded} \\
\delta_2(\lambda,(\tau,M)) & = & \lambda_y & \text{if } \lambda \text{ is } \Sigma\text{-guarded and } \tau =_\Sigma \lambda \\
\delta_2(\lambda,(\tau,M)) & = & \mathsf{false} & \text{if } \lambda \text{ is } \Sigma\text{-guarded and } \tau \neq_\Sigma \lambda \\
\delta_2(\lambda,(\tau,M)) & = & \lambda^?_y & \text{if } \lambda \text{ is not } \Sigma\text{-guarded} \\
\delta_2(\lambda_\uparrow,(\tau,M)) & = & \langle -1\rangle\lambda_y & \text{if } \lambda \text{ is } \Sigma\text{-guarded and } \tau =_\Sigma \lambda^- \\
\delta_2(\lambda_\uparrow,(\tau,M)) & = & \mathsf{false} & \text{if } \lambda \text{ is } \Sigma\text{-guarded and } \tau \neq_\Sigma \lambda^- \\
\delta_2(\lambda_\uparrow,(\tau,M)) & = & \lambda^?_y & \text{if } \lambda \text{ is not } \Sigma\text{-guarded}
\end{array}$$

where $\tau_y =_\Sigma t$ means that the atoms in τ that mention only y are identical to the Σ-relational atoms in t (up to renaming x to y), $\tau =_\Sigma \lambda$ means that the restriction of λ to Σ-atoms is exactly τ, and λ^- is obtained from λ by swapping x and y. We need further transitions to satisfy the obligations represented by states of the form t^0, which involves checking $A_\perp$-delimited bisimulations. Details are given in the appendix where also the correctness of the construction is proved.

▶ **Theorem 14.** *In GF^2, Σ-entailment and conservative extensions can be decided in time $2^{2^{p(|\varphi_2|\cdot\log|\varphi_1|)}}$, for some polynomial p. Moreover, Σ-inseparability is in* 2ExpTime.

Note that the time bound for conservative extensions given in Theorem 14 is double exponential only in the size of φ_2 (that is, the extension). In ontology engineering applications, φ_2 will often be small compared with φ_1.

A matching lower bound is proved using a reduction of the word problem of exponentially space-bounded alternating Turing machines, see the appendix for details. The construction is inspired by the proof from [16] that conservative extensions in the description logic $\mathcal{ALC}$ are 2ExpTime-hard, but the lower bound does not transfer directly since we are interested here in witness sentences that are formulated in GF^2 rather than in $\mathcal{ALC}$.

▶ **Theorem 15.** *In any fragment of FO that contains GF^2, the problems Σ-entailment, Σ-inseparability, conservative extensions, and strong Σ-entailment are* 2ExpTime*-hard.*

6 Conclusion

We have shown that conservative extensions are undecidable in (extensions of) GF and FO^2, and that they are decidable and 2ExpTime-complete in GF^2. It thus appears that decidability of conservative extensions is linked even more closely to the tree model property than decidability of the satisfiability problem: apart from cycles of length at most two, GF^2 enjoys a 'true' tree model property while GF only enjoys a bounded treewidth model property and FO^2 has a rather complex regular model property that is typically not even made explicit. As future work, it would be interesting to investigate whether conservative extensions remain decidable when guarded counting quantifiers, transitive relations, equivalence relations, or fixed points are added, see e.g. [35, 25, 23]. It would also be interesting to investigate a finite model version of conservative extensions.

References

1 Hajnal Andréka, István Németi, and Johan van Benthem. Modal languages and bounded fragments of predicate logic. *J. Philosophical Logic*, 27(3):217–274, 1998.
2 Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003. (2nd edition, 2007).
3 Vince Bárány, Georg Gottlob, and Martin Otto. Querying the guarded fragment. *Logical Methods in Computer Science*, 10(2), 2014.
4 Vince Bárány, Balder ten Cate, and Luc Segoufin. Guarded negation. *J. ACM*, 62(3):22:1–22:26, 2015.
5 Michael Benedikt, Balder ten Cate, and Michael Vanden Boom. Interpolation with decidable fixpoint logics. In *Proc. of LICS*, pages 378–389. IEEE Computer Society, 2015.
6 Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer, 1997.
7 Elena Botoeva, Boris Konev, Carsten Lutz, Vladislav Ryzhikov, Frank Wolter, and Michael Zakharyaschev. Inseparability and conservative extensions of description logic ontologies: A survey. In *Proc. of Reasoning Web*, volume 9885 of *LNCS*, pages 27–89. Springer, 2016.
8 Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *Journal of the ACM*, 28(1):114–133, 1981.
9 Chen Chung Chang and H. Jerome Keisler. *Model Theory*. Elsevier, 3rd edition, 1990.
10 Stephen D. Comer. Classes without the amalgamation property. *Pacific Journal of Mathematics*, 28:309–318, 1969.
11 Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research (JAIR)*, 31:273–318, 2008.
12 Giovanna D'Agostino and Marco Hollenberg. Logical questions concerning the μ-calculus: Interpolation, Lyndon and ł oś-Tarski. *J. Symb. Log.*, 65(1):310–332, 2000.
13 Giovanna D'Agostino and Giacomo Lenzi. Bisimulation quantifiers and uniform interpolation for guarded first order logic. *Theor. Comput. Sci.*, 563:75–85, 2015.
14 Răzvan Diaconescu, Joseph A. Goguen, and Petros Stefaneas. Logical support for modularisation. In Gerard Huet and Gordon Plotkin, editors, *Logical Environments*, pages 83–130, 1993.
15 E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *Proc. of FOCS*, pages 368–377. IEEE Computer Society, 1991.
16 Silvio Ghilardi, Carsten Lutz, and Frank Wolter. Did I damage my ontology? A case for conservative extensions in description logic. In *Proc. of KR*, pages 187–197. AAAI Press, 2006.
17 Valentin Goranko and Martin Otto. Model theory of modal logic. In Patrick Blackburn, Johan van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*, pages 249–330. Elsevier, 2006.
18 Michael J. C. Gordon and Thomas F. Melham, editors. *Introduction to HOL: A theorem proving environment for higher order logic*. Cambridge University Press, 1993.
19 Erich Grädel. On the restraining power of guards. *J. Symb. Log.*, 64(4):1719–1742, 1999.
20 Erich Grädel, Phokion G. Kolaitis, and Moshe Y. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3(1):53–69, 1997.
21 Erich Grädel and Martin Otto. The freedoms of (guarded) bisimulation. In Alexandru Baltag and Sonja Smets, editors, *Johan van Benthem on Logic and Information Dynamics*, pages 3–31. Springer, 2014.
22 Erich Grädel, Martin Otto, and Eric Rosen. Two-variable logic with counting is decidable. In *Proc. of LICS*, pages 306–317. IEEE Computer Society, 1997.

23 Erich Grädel and Igor Walukiewicz. Guarded fixed point logic. In *Proc. of LICS*, pages 45–54. IEEE Computer Society, 1999.

24 Eva Hoogland and Maarten Marx. Interpolation and definability in guarded fragments. *Studia Logica*, 70(3):373–409, 2002.

25 Emanuel Kieronski. On the complexity of the two-variable guarded fragment with transitive guards. *Inf. Comput.*, 204(11):1663–1703, 2006.

26 Emanuel Kieronski, Jakub Michaliszyn, Ian Pratt-Hartmann, and Lidia Tendera. Two-variable first-order logic with equivalence closure. *SIAM J. Comput.*, 43(3):1012–1063, 2014.

27 Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. Formal properties of modularisation. In Heiner Stuckenschmidt, Christine Parent, and Stefano Spaccapietra, editors, *Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization*, volume 5445 of *LNCS*, pages 25–66. Springer, 2009.

28 Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter. Model-theoretic inseparability and modularity of description logic ontologies. *Artificial Intelligence*, 203:66–103, 2013.

29 Carsten Lutz, Dirk Walther, and Frank Wolter. Conservative extensions in expressive description logics. In *IJCAI*, pages 453–458, 2007.

30 Carsten Lutz and Frank Wolter. Foundations for uniform interpolation and forgetting in expressive description logics. In *Proc. of IJCAI*, pages 989–995. IJCAI/AAAI, 2011.

31 Johannes Marti, Fatemeh Seifan, and Yde Venema. Uniform interpolation for coalgebraic fixpoint logic. In *CALCO*, volume 35 of *LIPIcs*, pages 238–252. Schloss Dagstuhl, 2015.

32 Michael Mortimer. On languages with two variables. *Math. Log. Q.*, 21(1):135–140, 1975.

33 Andrew M. Pitts. On an interpretation of second-order quantification in first-order intuitionistic propositional logic. *J. of Symbolic Logic*, 57, 1992.

34 Stephen Pollard. *Philosophical Introduction to Set Theory.* University of Notre Dame Press, 1990.

35 Ian Pratt-Hartmann. Complexity of the guarded two-variable fragment with counting quantifiers. *J. Log. Comput.*, 17(1):133–155, 2007.

36 Ian Pratt-Hartmann. Data-complexity of the two-variable fragment with counting quantifiers. *Inf. Comput.*, 207(8):867–888, 2009.

37 Dana Scott. A decision method for validity of sentences in two variables. *Journal of Symbolic Logic*, 27:1962, 1962.

38 Stephen G. Simpson. *Subsystems of Second Order Arithmetic.* Cambridge University Press, 2009.

39 Moshe Y. Vardi. Reasoning about the past with two-way automata. In *Proc. ICALP*, volume 1443 of *LNCS*, pages 628–641. Springer, 1998.

40 Albert Visser. Uniform interpolation and layered bisimulation. In *Gödel '96 (Brno, 1996)*, volume 6 of *Lecture Notes in Logic*, pages 139–164. Springer, 1996.

41 Thomas Wilke. Alternating tree automata, parity games, and modal μ-calculus. *Bulletin of the Belgian Mathematical Society*, 8(2), 2001.

A Proofs for Section 3

We split the proof of Lemma 6 into two parts.

▶ **Lemma 16.** *If M halts then $\varphi_1 \wedge \varphi_2$ is not a GF^2-conservative extension of φ_1.*

Proof. Assume that M halts. We define a witness ψ for non-conservativity. It says that every element participates in a substructure that represents the computation of M on input $(0,0)$, that is: if the computation is $(q_0,n_0,m_0),\ldots,(q_k,n_k,m_k)$, then there is an N-path of length k (but not longer) such that any object reachable in $i \leq k$ steps from the beginning of the path is labeled with q_i, has an outgoing R_0-path of length n_i and no longer outgoing R_0-path, and likewise for R_1 and m_i. In more detail, consider the Σ-structure $\mathfrak{A}$ with

$$A = \{0,\ldots,k\} \cup \{a^i_j \mid 0 < i \leq k, 0 < j < n_i\} \cup \{b^i_j \mid 0 < i \leq k, 0 < j < m_i\}$$

in which

$$\begin{array}{rcl} N^{\mathfrak{A}} & = & \{(i,i+1) \mid i < k\} \\ R_1^{\mathfrak{A}} & = & \bigcup_{i \leq k} \{(i,a^i_1),(a^i_1,a^i_2),\ldots,(a^i_{n_i-2},a^i_{n_i-1})\} \\ R_2^{\mathfrak{A}} & = & \bigcup_{i \leq k} \{(i,b^i_1),(b^i_1,b^i_2),\ldots,(b^i_{m_i-2},b^i_{m_i-1})\} \\ S^{\mathfrak{A}} & = & \{0\} \\ q^{\mathfrak{A}} & = & \{i \mid q_i = q\} \text{ for any } q \in Q. \end{array}$$

Then let ψ be a $GF^2(\Sigma)$-sentence that describes $\mathfrak{A}$ up to global $\text{GF}^2(\Sigma)$-bisimulations. Clearly $\mathfrak{A}$ satisfies $\varphi_1 \wedge \psi$. It thus remains to show that $\varphi_1 \wedge \varphi_2 \wedge \psi$ is unsatisfiable. But this is clear as there are no N-paths of length $> k$ in any model of ψ and since there are no defects in register updates in any model of ψ. ❑

▶ **Lemma 17.** *If there exists a Σ-structure that satisfies φ_1 and cannot be extended to a model of φ_2, then M halts.*

Proof. Let $\mathfrak{A}$ be a Σ-structure satisfying φ_1 that cannot be extended to a model of φ_2. Then $S^{\mathfrak{A}} \neq \emptyset$ and there exists an N-path labeled with states in Q starting in S. Since $\mathfrak{A}$ cannot be extended to a model of φ_2 the computation starting from S is finite. Moreover, one can readily prove by induction that no register update defects occur since otherwise φ_2 can be satisfied. ❑

We now prove Theorem 8 using a reduction of an undecidable tiling problem.

▶ **Definition 18.** A *tiling system* $\mathfrak{D} = (\mathfrak{T}, H, V, \mathsf{Right}, \mathsf{Left}, \mathsf{Top}, \mathsf{Bottom})$ consists of a finite set $\mathfrak{T}$ of *tiles*, horizontal and vertical *matching relations* $H, V \subseteq \mathfrak{T} \times \mathfrak{T}$, and subsets Right, Left, Top, and Bottom of $\mathfrak{T}$ containing the *right* tiles, *left* tiles, *top* tiles, and *bottom* tiles, respectively. A *solution* to $\mathfrak{D}$ is a triple (n,m,τ) where $n,m \in \mathbb{N}$ and $\tau : \{0,\ldots,n-1\} \times \{0,\ldots,m-1\} \rightarrow \mathfrak{T}$ such that the following hold:

1. $(\tau(i,j),\tau(i+1,j)) \in H$, for all $i < n$ and $j \leq m$;
2. $(\tau(i,j),\tau(i,j+1)) \in V$, for all $i \leq n$ and $j < m$;
3. $\tau(0,j) \in \mathsf{Left}$ and $\tau(n,j) \in \mathsf{Right}$, for all $j \leq m$;
4. $\tau(i,0) \in \mathsf{Bottom}$ and $\tau(i,m) \in \mathsf{Top}$, for all $i \leq n$.

We show how to convert a tiling system $\mathfrak{D}$ into FO^2-sentences φ_1 and φ_2 such that $\mathfrak{D}$ has a solution iff $\varphi_1 \wedge \varphi_2$ is not a conservative extension of φ_1. In particular, models of witness sentences will define solutions of $\mathfrak{D}$.

Let $\mathfrak{D} = (\mathfrak{T}, H, V, \mathsf{Right}, \mathsf{Left}, \mathsf{Top}, \mathsf{Bottom})$ be a tiling system. The formula φ_1 uses the following set Σ of predicates:

- binary predicates R_h and R_v (representing a grid) and S_h and S_v (for technical reasons),
- unary predicates T for every $T \in \mathfrak{T}$, G (for the domain of the grid), O (for the lower left corner of the grid), $B_\rightarrow$, $B_\leftarrow$, $B_\uparrow$, and $B_\downarrow$ (for the borders of the grid).

Then φ_1 is the conjunction of the following sentences:

1. Every position in the $n \times m$ grid is labeled with exactly one tile and the matching conditions are satisfied:
$$\forall x\big(Gx \rightarrow \bigvee_{T\in\mathfrak{T}} (Tx \wedge \bigwedge_{T'\in\mathfrak{T},\, T'\neq T} \neg T'x)\big)$$
$$\forall x\big(Gx \rightarrow \bigwedge_{T\in\mathfrak{T}} \big(Tx \rightarrow (\bigvee_{(T,T')\in H} \forall y(R_hxy \rightarrow T'y) \wedge \bigvee_{(T,T')\in V} \forall y(R_vxy \rightarrow T'y))\big)\big).$$
2. The predicates $B_\rightarrow$, $B_\leftarrow$, $B_\uparrow$, and $B_\downarrow$ mark the borders of the grid:
$$\forall x\big(Gx \wedge B_\rightarrow x \rightarrow \big(\neg\exists y R_hxy \wedge \forall y(R_vxy \rightarrow B_\rightarrow y) \wedge \forall y(R_vyx \rightarrow B_\rightarrow y)\big)\big)$$
$$\forall x\big(Gx \wedge \neg B_\rightarrow x \rightarrow \exists y R_hxy\big)$$
and similarly for $B_\leftarrow$, $B_\uparrow$, and $B_\downarrow$.
3. There is a grid origin:
$$\exists x\,(Ox \wedge B_\leftarrow x \wedge B_\downarrow x).$$
4. The grid elements are marked by G:
$$\forall x\big(Ox \rightarrow Gx\big), \quad \forall x(Gx \rightarrow \forall y(R_hxy \rightarrow Gy)), \quad \forall x(Gx \rightarrow \forall y(R_vxy \rightarrow Gy)).$$
5. The tiles on border positions are labeled with appropriate tiles:
$$\forall x(B_\rightarrow x \rightarrow \bigvee_{T\in\mathsf{Right}} T(x)).$$
and similarly for $B_\leftarrow$, $B_\uparrow$, and $B_\downarrow$.
6. The predicates S_h and S_v occur in φ_1: any FO^2-tautology using them.

This finishes the definition of φ_1. The sentence φ_2 introduces two new unary predicates Q and P and is the conjuntion of $\exists x(Ox \wedge Qx)$ and

$$\forall x\big(Qx \rightarrow \big(\exists y(R_hxy \wedge Qy) \vee \exists y(R_vxy \wedge Qy) \vee \varphi_D x\big)\big)$$

where

$$\varphi_D x = \exists y\big(R_hxy \wedge \forall x(R_vyx \rightarrow Px)\big) \wedge \exists y\big(R_vxy \wedge \forall x(R_hyx \rightarrow \neg Px)\big)$$

Thus, φ_D describes a defect in the grid: there exist an R_h-successor y_1 and an R_v-successor y_2 of x such that every R_v-successor of y_1 is labeled with P and every R_h-successor of y_2 is labeled with $\neg P$. Informally, we can satisfy φ_2 only if from some element of O, there is an infinite R_h/R_v-path or a non-closing grid cell can be reached by a finite such path. To make this precise, we introduce some notation. Let $\Sigma' \supseteq \Sigma$ and let $\mathfrak{B}$ be a Σ'-structure. Then the Σ-structure $\mathfrak{A}$ obtained from $\mathfrak{B}$ by removing the interpretation of predicates in $\Sigma' \setminus \Sigma$ is called the *Σ-reduct of* $\mathfrak{B}$ and $\mathfrak{B}$ is called a *$\Sigma' \setminus \Sigma$-extension of* $\mathfrak{A}$. For a Σ-structure $\mathfrak{A}$, we say that *$a \in A$ is the root of a non-closing grid cell* if there are $(a, b_1) \in R_h^{\mathfrak{A}}$ and $(a, b_2) \in R_v^{\mathfrak{A}}$ such that there does not exist a $c \in A$ with $(b_1, c) \in R_v^{\mathfrak{A}}$ and $(b_2, c) \in R_h^{\mathfrak{A}}$. Now, it is straightforward to show the following characterization of φ_2.

▶ **Lemma 19.** *φ_2 can be satisfied in a $\{Q, P\}$-extension of a Σ-structure $\mathfrak{A}$ iff there exists an element of $O^{\mathfrak{A}}$ that starts an infinite R_h/R_v-path or a finite R_h/R_v-path to a root of a non-closing grid cell.*

We now argue that $\mathfrak{D}$ has a solution iff $\varphi_1 \wedge \varphi_2$ is not a conservative extension of φ_1.

▶ **Lemma 20.** *If $\mathfrak{D}$ has a solution then $\varphi_1 \wedge \varphi_2$ is not a FO^2-conservative extension of φ_1.*

Proof. Assume that $\mathfrak{D}$ has a solution (n, m, τ). We define a witness ψ, first using additional fresh unary predicates and then argueing that these can be removed. Thus introduce fresh unary predicates $P_{i,j}$ for all $i < n$ and $j < m$. Intuitively, $P_{i,j}$ identifies grid position (i, j). Set

$$\begin{array}{rcl}\psi & = & \forall x\,(Gx \rightarrow \bigvee_{i,j} P_{i,j}x) \\ & & \wedge \displaystyle\bigwedge_{(i,j)\neq(i',j')} \forall x\,\neg(P_{i,j}x \wedge P_{i',j'}x) \\ & & \wedge\ \forall x \forall y\,(R_h xy \leftrightarrow \bigvee_{i,j} P_{i,j}x \wedge P_{i+1,j}y) \\ & & \wedge\ \forall x \forall y\,(R_v x, y \leftrightarrow \bigvee_{i,j} P_{i,j}x \wedge P_{i,j+1}y) \\ & & \wedge\ \forall x\,(Ox \rightarrow P_{0,0}x).\end{array}$$

We first show that $\varphi_1 \wedge \psi$ is satisfiable. As the model, simply take the standard $n \times m$-grid in which all positions are labeled with $P_{i,j}$, G, O, $B_\rightarrow$ etc in the expected way, and that is tiled according to τ. It is easily verified that this structure satisfies both φ_1 and ψ. It thus remains to show that $\varphi_1 \wedge \varphi_2 \wedge \psi$ is unsatisfiable. By Lemma 19 it suffices to show that there is no model $\mathfrak{A}$ of $\varphi_1 \wedge \psi$ in which there exists an element of $O^\mathfrak{A}$ starting an infinite R_h/R_v-path or a finite R_h/R_v-path leading to a root of a non-closing grid cell. Assume for a proof by contradiction that there exists such a model $\mathfrak{A}$ and $a \in O^\mathfrak{A}$. Then we find a sequence $a_0 R^\mathfrak{A}_{z_0} a_1 R^\mathfrak{A}_{z_1} a_2 \cdots$ with $a_0 = a$ and $z_i \in \{x, y\}$ such that either some a_h is the root of a non-closing grid cell or the sequence is infinite. By φ_1 and the first conjunct of ψ for each a_k there exists $P_{i,j}$ with $a_k \in P^\mathfrak{A}_{i,j}$. By the last conjunct of ψ, $a_0 \in P^\mathfrak{A}_{0,0}$. By the remaining conjuncts of ψ we have $k \geq i + j$ if $a_k \in P^\mathfrak{A}_{i,j}$ and it follows that there is no a_k with $k > n + m$. Thus, assume some a_k is the root of a non-closing grid. Then we have $(a_k, b_1) \in R^\mathfrak{A}_h$ and $(a_k, b_2) \in R^\mathfrak{A}_v$ such that there is no $c \in A$ with $(b_1, c) \in R^\mathfrak{A}_v$ and $(b_2, c) \in R^\mathfrak{A}_h$. By ψ, there are i, j with $b_1 \in P^\mathfrak{A}_{i+1,j}$ and $b_2 \in P^\mathfrak{A}_{i,j+1}$. By the second set of conjuncts of φ_1 there exists $(b_1, c) \in R^\mathfrak{A}_v$. By ψ, $c \in P^\mathfrak{A}_{i+1,j+1}$. But then again using ψ we obtain that $(b_2, c) \in R^\mathfrak{A}_h$ and we have derived a contradiction.

As announced, we now show how to remove the additional predicates $P_{i,j}$. To this end, we use the binary predicates S_h, S_v. In the sentence ψ, we replace every occurrence of $P_{i,j}(x)$ with a formula saying: there is an outgoing S_h-path of length i, but not of length $i+1$ and an outgoing S_v-path of length j, but not of length $j + 1$. When we build a model of $\varphi_1 \wedge \psi$, we now need to introduce additional elements for the "counting paths". We make the predicate G false on all those elements and true everywhere on the grid. ❑

▶ **Lemma 21.** *If there exists a Σ-structure that satisfies φ_1 and cannot be extended to a model of φ_2, then $\mathfrak{D}$ has solution.*

Proof. Take a Σ-structure $\mathfrak{A}$ satisfying φ_1 that cannot be extended to a model of φ_2. By the conjunct of φ_1 given in Item 3, $O^\mathfrak{A} \cap B^\mathfrak{A}_\leftarrow \cap B^\mathfrak{A}_\downarrow \neq \emptyset$. Take $a \in O^\mathfrak{A} \cap B^\mathfrak{A}_\leftarrow \cap B^\mathfrak{A}_\downarrow$. By Lemma 19, a does not start an infinite R_h/R_v-path or a finite R_h/R_v-path leading to the root of a non-closing grid cell. Using the conjuncts of φ_1 it is now straightforward to read off a tiling from $\mathfrak{A}$. ❑

Theorem 8 is an immediate consequence of Lemmas 20 and 21.

Note that the sentences φ_1 and φ_2 can be replaced by equivalent $\mathcal{ALC}$-TBoxes: in φ_2, we can replace the conjunct $\exists x(Ox \wedge Qx)$ which cannot be expressed in $\mathcal{ALC}$ by the concept inclusion $\top \sqsubseteq \exists S.(O \sqcap Q)$ with S a fresh binary predicate. Consequently, $\mathrm{FO}^2(\Sigma)$-inseparability of $\mathcal{ALC}$-TBoxes is undecidable.

B Preliminaries on Bisimulations

We first show that $\mathrm{GF}^2(\Sigma)$-bisimulations characterize the expressive power of $\mathrm{GF}^2(\Sigma)$. The proofs are standard [21, 17, 1]. We start by giving a precise definition of k-$\mathrm{GF}^2(\Sigma)$ bisimilarity. Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures, $a \in A$, and $b \in B$. The definition is by induction on $k \geq 0$. Then $(\mathfrak{A}, a) \sim_\Sigma^0 (\mathfrak{B}, b)$ iff $\mathrm{at}_\mathfrak{A}^\Sigma(a) = \mathrm{at}_\mathfrak{B}^\Sigma(b)$ and $(\mathfrak{A}, a) \sim_\Sigma^{k+1} (\mathfrak{B}, b)$ iff $\mathrm{at}_\mathfrak{A}^\Sigma(a) = \mathrm{at}_\mathfrak{B}^\Sigma(b)$ and

1. for every $a' \neq a$ such that $\mathrm{at}_\mathfrak{A}^\Sigma(a, a')$ is guarded there exists $b' \neq b$ such that $\mathrm{at}_\mathfrak{A}^\Sigma(a, a') = \mathrm{at}_\mathfrak{B}^\Sigma(b, b')$ and $(\mathfrak{A}, a') \sim_\Sigma^k (\mathfrak{B}, b')$
2. for every $b' \neq b$ such that $\mathrm{at}_\mathfrak{B}^\Sigma(b, b')$ is guarded there exists $a' \neq a$ such that $\mathrm{at}_\mathfrak{B}^\Sigma(b, b') = \mathrm{at}_\mathfrak{A}^\Sigma(a, a')$ and $(\mathfrak{A}, a') \sim_\Sigma^k (\mathfrak{B}, b')$.

Denote by openGF^2 the fragment of GF^2 consisting of all GF^2 formulas with one free variable in which equality is not used as a guard and which do not contain a subformula that is a sentence. It is not difficult to prove the following result.

▶ **Lemma 22.** *Every GF^2 sentence is equivalent to a Boolean combination of sentences of the form $\forall x \varphi(x)$, where $\varphi(x)$ is a $openGF^2$ formula.*

A structure $\mathfrak{A}$ is *ω-saturated* if for every finite set $\{a_1, \ldots, a_n\} \subseteq A$ and every set $\Gamma(x)$ of FO formulas using elements of $\{a_1, \ldots, a_n\}$ as constants the following holds: if every finite subset of $\Gamma(x)$ is satisfiable in the structure $(\mathfrak{A}, a_1, \ldots, a_n)$, then $\Gamma(x)$ is satisfiable in $(\mathfrak{A}, a_1, \ldots, a_n)$. For every structure $\mathfrak{A}$ there exists an elementary extension $\mathfrak{A}'$ of $\mathfrak{A}$ that is ω-saturated [9]. Mostly we only require a weaker form of saturation. A structure $\mathfrak{A}$ is *successor-saturated* if for any $a \in A$ and set $\Gamma(x)$ of openGF^2 formulas the following holds for any atomic guarded binary type τ: if for any finite subset Γ' of Γ there exists $a' \neq a$ with $\mathrm{at}_\mathfrak{A}(a, a') = \tau$ and $\mathfrak{A} \models \psi(a')$ for all $\psi \in \Gamma'$, then there exists $b' \neq a$ with $\mathrm{at}_\mathfrak{A}(a, b') = \tau$ and $\mathfrak{A} \models \psi(b')$ for all $\psi \in \Gamma$. Observe that structures of finite outdegree and ω-saturated structures are successor-saturated.

The *depth* of a GF^2 formula φ is the number of nestings of guarded quantifications in φ. We first characterize openGF^2. The proof is standard and omitted.

▶ **Lemma 23.** *Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures, Σ a signature, and $a \in A$, $b \in B$.*

1. *The following conditions are equivalent for all $k \geq 0$:*
   - *$\mathfrak{A} \models \varphi(a)$ iff $\mathfrak{B} \models \varphi(b)$ holds for all $openGF^2(\Sigma)$ formulas $\varphi(x)$ of depth k;*
   - *$(\mathfrak{A}, a) \sim_\Sigma^k (\mathfrak{B}, b)$.*
2. *If $(\mathfrak{A}, a) \sim_\Sigma (\mathfrak{B}, b)$, then $\mathfrak{A} \models \varphi(a)$ iff $\mathfrak{B} \models \varphi(b)$ holds for all $openGF^2(\Sigma)$ formulas $\varphi(x)$. The converse direction holds if $\mathfrak{A}$ and $\mathfrak{B}$ are successor-saturated.*

We also require the following link between bounded bisimulations and unbounded bisimulations which follows from Lemma 23.

▶ **Lemma 24.** *Let $\mathfrak{A}$ and $\mathfrak{B}$ be successor-saturated structures, $a \in A$, and $b \in B$. If $(\mathfrak{A}, a) \sim_\Sigma^k (\mathfrak{B}, b)$ for all $k \geq 0$, then $(\mathfrak{A}, a) \sim_\Sigma (\mathfrak{B}, b)$.*

We now consider 'global' versions of the bounded bisimulations introduced above to characterize GF^2. Call structures $\mathfrak{A}$ and $\mathfrak{B}$ *globally k-$GF^2(\Sigma)$-bisimilar* if for all $a \in A$ there exists $b \in B$ such that $(\mathfrak{A}, a) \sim_\Sigma^k (\mathfrak{B}, b)$ and, conversely, for every $b \in B$ there exists $a \in A$ with $(\mathfrak{A}, a) \sim_\Sigma^k (\mathfrak{B}, b)$. $\mathfrak{A}$ and $\mathfrak{B}$ are *globally finitely $GF^2(\Sigma)$-bisimilar* iff they are globally k-$\mathrm{GF}^2(\Sigma)$-bisimilar for all $k \geq 0$. The following characterization result now follows from Lemma 22 and Lemma 23.

▶ **Lemma 25.** *Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures and Σ a signature.*

1. *The following conditions are equivalent:*
   - $\mathfrak{A} \models \varphi$ *iff* $\mathfrak{B} \models \varphi$ *holds for all* $GF^2(\Sigma)$ *sentences* φ*;*
   - $\mathfrak{A}$ *and* $\mathfrak{B}$ *are globally finitely* $GF^2(\Sigma)$*-bisimilar.*
2. *If* $\mathfrak{A}$ *and* $\mathfrak{B}$ *are globally* $GF^2(\Sigma)$*-bisimilar, then* $\mathfrak{A} \models \varphi$ *iff* $\mathfrak{B} \models \varphi$ *holds for all* $GF^2(\Sigma)$ *sentences* φ*. The converse direction holds if* $\mathfrak{A}$ *and* $\mathfrak{B}$ *are* ω*-saturated.*

Observe that in Lemma 25 we cannot replace ω-saturation by successor-saturation or finite outdegree.

C Proofs for Section 4

Based on the results presented in the previous section we prove the following characterization of Σ-entailment in FO^2.

Theorem 11 *Let* φ_1, φ_2 *be* GF^2*-sentences and* Σ *a signature. Then* $\varphi_1 \models_\Sigma \varphi_2$ *iff for every model* $\mathfrak{A}$ *of* φ_1 *of finite outdegree, there is a model* $\mathfrak{B}$ *of* φ_2 *such that*
1. *for every* $a \in A$ *there is a* $b \in B$ *such that* $(\mathfrak{A}, a) \sim_\Sigma (\mathfrak{B}, b)$
2. *for every* $b \in B$ *and every* $k \geq 0$*, there is an* $a \in A$ *such that* $(\mathfrak{A}, a) \sim_\Sigma^k (\mathfrak{B}, b)$*.*

Proof. "if". Assume that for every model $\mathfrak{A}$ of φ_1 of finite outdegree, there is a model $\mathfrak{B}$ of φ_2 as described in Theorem 11. Take a Σ-sentence ψ such that $\varphi_1 \wedge \psi$ is satisfiable. We have to show that $\varphi_2 \wedge \psi$ is satisfiable. We find a model $\mathfrak{A}$ of $\varphi_1 \wedge \psi$ that has finite outdegree. By assumption, there is a model $\mathfrak{B}$ of φ_2 that satisfies Conditions 1 and 2 of Theorem 11. It suffices to show that $\mathfrak{B}$ satisfies ψ. But this follows from Lemma 25.

"only if". Assume that $\varphi_1 \models_\Sigma \varphi_2$. Let $\mathfrak{A}$ be a model of φ_1 of finite outdegree. Let Γ denote the set of all $\text{GF}^2(\Sigma)$ sentences ψ with $\mathfrak{A} \models \psi$. Then $\varphi_1 \wedge \bigwedge \Gamma'$ is satisfiable for every finite subset Γ' of Γ. As $\varphi_1 \models_\Sigma \varphi_2$, $\varphi_2 \wedge \bigwedge \Gamma'$ is satisfiable for every finite subset Γ' of Γ. By compactness $\{\varphi_2\} \cup \Gamma$ is satisfiable. Then there exists an ω-saturated model $\mathfrak{B}$ of $\{\varphi_2\} \cup \Gamma$. By ω-saturatedness, for every $a \in A$ there exists $b \in B$ such that $\mathfrak{A} \models \varphi(a)$ iff $\mathfrak{B} \models \varphi(b)$ holds for all formulas $\varphi(x)$ in $\text{openGF}^2(\Sigma)$. By Lemma 23, we have $(\mathfrak{A}, a) \sim_\Sigma (\mathfrak{B}, b)$, as required for Condition 1. Condition 2 follows from Lemma 23. ❏

Before we come to the proof of Theorem 12 we prove another characterization of Σ-entailment in GF^2. If $\mathfrak{A}$ is a forest structure with $a, a' \in A$, then we write $a \prec a'$ iff a and a' are part of the same Σ-tree in $\mathfrak{A}$ and a is a ancestor of a' (recall that a Σ-tree in a forest structure $\mathfrak{A}$ is a maximal Σ-connected substructure of $\mathfrak{A}$ and that we always assume a fixed root in trees within forest structures). For $\mathfrak{A}$ and $\mathfrak{B}$ structures and $a_\perp \in A$, an *$a_\perp$-delimited $GF^2(\Sigma)$-bisimulation between $\mathfrak{A}$ and $\mathfrak{B}$* is defined like a $\text{GF}^2(\Sigma)$-bisimulation except that Conditions 2 and 3 are not required to hold when $a = a_\perp$. We indicate the existence of an $a_\perp$-delimited bisimulation by writing $(\mathfrak{A}, a) \sim_\Sigma^{a_\perp} (\mathfrak{B}, b)$. This requires $a_\perp \preceq a$. We now give a characterization of Σ-entailment using forest models in which we replace the bounded backward condition by an unbounded condition.

▶ **Theorem 26.** *Let* φ_1, φ_2 *be* GF^2*-sentences and* Σ *a signature. Then* $\varphi_1 \models_\Sigma \varphi_2$ *iff for every regular forest model* $\mathfrak{A}$ *of* φ_1 *that has finite outdegree there is a model* $\mathfrak{B}$ *of* φ_2 *such that*
1. *for every* $a \in A$ *there is a* $b \in B$ *such that* $(\mathfrak{A}, a) \sim_\Sigma (\mathfrak{B}, b)$
2. *for every* $b \in B$*, one of the following holds:*
   a. *there is an* $a \in A$ *such that* $(\mathfrak{A}, a) \sim_\Sigma (\mathfrak{B}, b)$*;*
   b. *there are* $a_\perp, a_0, a_1, \dots, a'_0, a'_1, \dots \in A$ *such that* $a_\perp \prec a_0 \prec a_1 \prec \cdots$ *and, for all* $i \geq 0$*,* $a_i \prec a'_i$ *and* $(\mathfrak{A}, a'_i) \sim_\Sigma^{a_\perp} (\mathfrak{B}, b)$*.*

Proof. Using the proof of Theorem 11, the fact that every (successor-saturated/finite outdegree) structure $\mathfrak{A}$ can be unfolded into a globally $\mathrm{GF}^2(\Sigma)$-bisimilar (successor saturated/finite outdegree) forest model $\mathfrak{B}$, and the fact that, consequently, every satisfiable GF^2 formula is satisfiable in a regular forest model of finite outdegree one can easily prove the following variant of Theorem 11 based on forest models:

Fact 1. Let φ_1, φ_2 be GF^2-sentences and Σ a signature. Then $\varphi_1 \models_\Sigma \varphi_2$ iff for every regular forest model $\mathfrak{A}$ of φ_1 that has finite outdegree there is a (successor saturated) forest model $\mathfrak{B}$ of φ_2 such that

1. for every $a \in A$ there is a $b \in B$ such that $(\mathfrak{A}, a) \sim_\Sigma (\mathfrak{B}, b)$
2. for every $b \in B$ and every $k \geq 0$, there is an $a \in A$ such that $(\mathfrak{A}, a) \sim_\Sigma^k (\mathfrak{B}, b)$.

To show Theorem 26 it therefore suffices to show that for every regular forest model $\mathfrak{A}$ of φ_1 that has finite outdegree and every successor-saturated forest model $\mathfrak{B}$ of φ_2, Condition 2 in Fact 1 is equivalent to Condition 2 of Theorem 26.

Thus, let $\mathfrak{A}$ and $\mathfrak{B}$ be as described. The interesting direction is to prove that if Condition 2 in Fact 1 holds then Condition 2 of Theorem 26 holds. Thus, assume that Condition 2 in Fact 1 holds. Take $b \in B$. We may assume it is a root b of a Σ-tree in $\mathfrak{B}$. Then there are $a_0, a_1, \dots \in A$ such that for all k, $(\mathfrak{A}, a_k) \sim_\Sigma^k (\mathfrak{B}, b)$. If infinitely many of the a_i are identical, then there is an $a \in A$ such that $(\mathfrak{A}, a) \sim_\Sigma^k (\mathfrak{B}, b)$ for all $k \geq 0$, thus $(\mathfrak{A}, a) \sim_\Sigma (\mathfrak{B}, b)$ by Lemma 24 and we are done. Therefore, assume that there are infinitely many distinct a_i. By 'skipping' elements in the sequence $a_0, a_1, \dots$, we can then achieve that the a_i are all distinct.

Two nodes $a, a' \in A$ are *downwards isomorphic*, written $a \sim_\downarrow a'$, if they are the roots of isomorphic subtrees. For a forest structure $\mathfrak{A}$, $a \in A$, and $i \geq 0$, we denote by $\mathfrak{A}|_a^{\uparrow i}$ the path structure obtained by restricting $\mathfrak{A}$ to those elements that can be reached from a by traveling at most i steps towards the root of the tree in $\mathfrak{A}$ that a is part of (including a itself). For $a, a' \in A$ and $i \geq 0$, we write $a \approx_i a'$ if there is an isomorphism ι from $\mathfrak{A}|_a^{\uparrow i}$ to $\mathfrak{A}|_{a'}^{\uparrow i}$ with $\iota(a) = a'$ such that $c \sim_\downarrow \iota(c)$ for all c. Since $\mathfrak{A}$ is regular, $\mathfrak{A}$ contains only finitely many equivalence classes for each $\approx_i$. By skipping a_i's, we can thus achieve that

$(*)$ $a_i \approx_k a_j$ for all i, k, j with $k \leq i$ and $j > i$.

This also implies that each a_i is at least i steps away from the root of the tree in $\mathfrak{A}$ that it is in (since there are infinitely many a_i, they must be unboundedly deep in their respective tree, and it remains to apply $(*)$). Let c_i denote the element of A reached from a_i by traveling i steps towards the root. Since $\mathfrak{A}$ is regular, there must be an infinite subsequence $a_{\ell_0}, a_{\ell_1}, \dots$ of $a_0, a_1, \dots$ such that $c_{\ell_i} \sim_\downarrow c_{\ell_j}$ for all i, j.

Choose some $a_\perp \in A$ with $a_\perp \sim_\downarrow c_{\ell_i}$ for all i (equivalently: for some i). We can assume w.l.o.g. that each a_{ℓ_i} is in the subtree rooted at $a_\perp$ and that when traveling ℓ_i steps from a_{ℓ_i} towards the root of the subtree that $a_\perp$ is in, then we reach exactly $a_\perp$.

Let $\mathfrak{A}^*$ be the structure obtained in the limit of the neighborhoods $\mathfrak{A}|^0_{a_{\ell_0}}, \mathcal{A}|^1_{a_{\ell_1}}, \dots$. That is, we start with the subtree of $\mathfrak{A}$ rooted at a_{ℓ_0}, renaming a_{ℓ_0} to a^*, and then proceed as follows: after the i-th step, the constructed structure is isomorphic to the subtree of $\mathfrak{A}$ rooted at $a_\perp$ via an isomorphism that maps a^* to a_{ℓ_i} and the root to $a_\perp$; by $(*)$, we can thus add a path of predecessor to the root of the structure constructed so far, and then add additional subtrees to the nodes on the path as additional successors, making sure that the obtained structure is isomorphic to the subtree of $\mathfrak{A}$ rooted at $a_\perp$ via an isomorphism that maps a^* to $a_{\ell_{i+1}}$ and the new root to $a_\perp$. By construction, $(\mathfrak{A}^*, a^*) \sim_\Sigma^k (\mathfrak{B}, b)$ for all $k \geq 0$ and thus Lemma 24 yields $(\mathfrak{A}^*, a^*) \sim_\Sigma (\mathfrak{B}, b)$.

Take some a_{ℓ_i}. We aim to show that $(\mathfrak{A}, a_{\ell_i}) \sim_\Sigma^{a_\perp} (\mathfrak{B}, b)$. Let c be the element reached from a^* in $\mathfrak{A}^*$ by traveling ℓ_i steps upwards and recall that $a_\perp$ is the element reached from a_{ℓ_i} in $\mathfrak{A}$ by traveling ℓ_i steps upwards. By construction of $\mathfrak{A}^*$, we find an isomorphism from the subtree in $\mathfrak{A}^*$ rooted at c to the subtree in $\mathfrak{A}$ rooted at $a_\perp$ that takes c to $a_\perp$ and a^* to a_i. From $(\mathfrak{A}^*, a^*) \sim_\Sigma (\mathfrak{B}, b)$, we thus obtain the desired $a_\perp$-delimited Σ-bisimulation that witnesses $(\mathfrak{A}, a_i) \sim_\Sigma^{a_\perp} (\mathfrak{B}, b)$.

It remains to show the existence of the required elements $a_0', a_1', \ldots$, that is, to show that there is a path through the subtree of $\mathfrak{A}$ rooted at $a_\perp$ such that each a_{ℓ_i} is either on the path or can be reached by branching off at a different point of the path. This can be done in the following straightforward way. Starting at $a_\perp$, we define the path step by step. In every step, there must be at least one successor which is the root of a subtree that contains infinitely many a_{ℓ_i}'s since $\mathfrak{A}$ has finite outdegree. We always proceed by choosing such a successor. This almost achieves the desired result, except that not all a_{ℓ_i} are reachable from a *distinct* node on the path by traveling downwards. However, there are infinitely many nodes on the path from which at least one a_{ℓ_i} can be reached by traveling downwards, so the problem can be cured by skipping a_{ℓ_i}'s. ❏

We are now in a position to prove Theorem 12. We require the following extended version of k-GF^2-bisimilarity which respects the successor relation in forest structures. Let $\mathfrak{A}$ and $\mathfrak{B}$ be forest structures, $a \in A$, and $b \in B$. The definition is by induction on $k \geq 0$. Then $(\mathfrak{A}, a) \sim_\Sigma^{0,\text{succ}} (\mathfrak{B}, b)$ iff $\text{at}_{\mathfrak{A}}^\Sigma(a) = \text{at}_{\mathfrak{B}}^\Sigma(b)$ and $(\mathfrak{A}, a) \sim_\Sigma^{k+1,\text{succ}} (\mathfrak{B}, b)$ iff $\text{at}_{\mathfrak{A}}^\Sigma(a) = \text{at}_{\mathfrak{B}}^\Sigma(b)$ and

1. for every $a' \neq a$ such that $\text{at}_{\mathfrak{A}}^\Sigma(a, a')$ is guarded there exists $b' \neq b$ such that $\text{at}_{\mathfrak{A}}^\Sigma(a, a') = \text{at}_{\mathfrak{B}}^\Sigma(b, b')$ and b' is a successor of b in $\mathfrak{B}$ iff a' is a successor of a in $\mathfrak{A}$ and $(\mathfrak{A}, a') \sim_\Sigma^{k,\text{succ}} (\mathfrak{B}, b')$
2. for every $b' \neq b$ such that $\text{at}_{\mathfrak{A}}^\Sigma(b, b')$ is guarded there exists $a' \neq a$ such that $\text{at}_{\mathfrak{A}}^\Sigma(b, b') = \text{at}_{\mathfrak{B}}^\Sigma(a, a')$ and a' is a successor of a in $\mathfrak{A}$ iff b' is a successor of b in $\mathfrak{B}$ and $(\mathfrak{A}, a') \sim_\Sigma^{k,\text{succ}} (\mathfrak{B}, b')$.

Theorem 12 *Let φ_1, φ_2 be GF^2-sentences and Σ a signature. Then $\varphi_1 \models_\Sigma \varphi_2$ iff for every regular forest model $\mathfrak{A}$ of φ_1 that has finite outdegree and every set $A_\perp \subseteq A$ with $A_\perp \cap \rho$ infinite for any infinite Σ-path ρ in $\mathfrak{A}$ there is a model $\mathfrak{B}$ of φ_2 such that*

1. *for every $a \in A$, there is a $b \in B$ such that $(\mathfrak{A}, a) \sim_\Sigma (\mathfrak{B}, b)$*
2. *for every 1-type t for φ_2 that is realized in $\mathfrak{B}$, there are $a \in A$ and $b \in B$ such that $\mathsf{tp}_{\mathfrak{B}}(b) = t$ and $(\mathfrak{A}, a) \sim_\Sigma^{A_\perp} (\mathfrak{B}, b)$.*

Proof. ($\Leftarrow$) It suffices to show that for every $m > 0$ and every regular forest model $\mathfrak{A}$ of φ_1 that has finite outdegree there exists a model $\mathfrak{B}$ of φ_2 such that $\mathfrak{A}$ and $\mathfrak{B}$ are globally m-GF$^2(\Sigma)$-bisimilar. Assume $m > 0$ and a regular forest model $\mathfrak{A}$ of φ_1 that has finite outdegree is given. Let m' be the maximum of m and the guarded quantifier depth of φ_2. Then $f(m, \varphi_2)$ denotes the maximal number of nodes in any $\Sigma \cup \mathsf{sig}(\varphi_2)$-forest model $\mathfrak{C}$ which are pairwise $\sim^{m',\text{succ}}$-incomparable. Define $A_\perp \subseteq A$ on every Σ-tree with root r in $\mathfrak{A}$ in such a way that $a \in A_\perp$ iff the distance between r and a is $kf(m, \varphi_2)$ for some $k \geq 0$. Let $\mathfrak{B}$ be a forest shaped model of φ_2 satisfying the conditions of Theorem 12. One can easily modify $\mathfrak{B}$ in such a way that in addition to the conditions given in the theorem

($*$) every 1-type t for φ_2 that is realized in $\mathfrak{B}$ is realized in the root of a Σ-tree in $\mathfrak{B}$ and for every root r of a Σ-tree in $\mathfrak{B}$ there exists $a \in A$ such that $(\mathfrak{A}, a) \sim_\Sigma^{A_\perp} (\mathfrak{B}, r)$.

To show ($*$) first pick for every $a \in A$ a $b \in B$ with $(\mathfrak{A}, a) \sim_\Sigma (\mathfrak{B}, b)$. Let S_1 be the set of b's just picked and let $\mathfrak{B}_1$ be the disjoint union of the structures induced in $\mathfrak{B}$ by the Σ-trees whose roots are in S_1. Next pick for every 1-type t for φ_2 that is realized in $\mathfrak{B}$ a $b \in B$

that realizes t. Let S_2 be the set of b's just picked and let $\mathfrak{B}_2$ be the disjoint union of the structures induced in $\mathfrak{B}$ by the Σ-trees whose roots are in S_2. Finally, we add (recursively) witnesses for guarded existential quantifiers not involving binary predicates from Σ to the disjoint union $\mathfrak{B}'$ of $\mathfrak{B}_1$ and $\mathfrak{B}_2$. In detail, take for any b in $\mathfrak{B}'$ its copy b' in $\mathfrak{B}$ and assume c' in $\mathfrak{B}$ is such that $\{R \mid (b',c') \in R^{\mathfrak{B}}$ or $(c',b') \in R^{\mathfrak{B}}\}$ is non-empty and contains no predicate in Σ. Then add to $\mathfrak{B}'$ a copy of the Σ-tree in $\mathfrak{B}'$ whose root c realizes the same 1-type for φ_2 as c' and connect c to b by adding for all binary predicates R the pair (b,c) to the extension of R if $(b',c') \in R^{\mathfrak{B}}$ and the pair (c,b) to the extension of R if $(c',b') \in R^{\mathfrak{B}}$. We apply this procedure recursively to the new structure (in a fair way) and obtain the desired structure as the limit of the resulting sequence of structures.

We now modify $\mathfrak{B}$ in such a way that the resulting structure is still a model of φ_2 but in addition globally m-GF$^2(\Sigma)$-bisimilar to $\mathfrak{A}$. Consider the structure $\mathfrak{B}_r$ induced by the Σ-tree with root r in $\mathfrak{B}$. If there exists an $a \in A$ with $(\mathfrak{A},a) \sim_\Sigma (\mathfrak{B},r)$ then we do not modify $\mathfrak{B}_r$ and set $\mathfrak{B}_r^u = \mathfrak{B}_r$. If no such a exists, then we modify $\mathfrak{B}_r$ in such a way that every b in the resulting Σ-tree is m-GF$^2(\Sigma)$-bisimilar to some $a \in A$. Note that we only know that there exists $a \in A$ such that $(\mathfrak{A},a) \sim_\Sigma^{A_\perp} (\mathfrak{B},r)$. By construction of $A_\perp$ this implies that $(\mathfrak{A},a)$ and $(\mathfrak{B},r)$ are $f(m,\varphi_2)$-GF$^2(\Sigma)$-bisimilar. Thus, it suffices to modify $\mathfrak{B}_r$ in such a way that every node b in the Σ-tree becomes m-GF$^2(\Sigma)$-bisimilar to some b' in the original $\mathfrak{B}_r$ with distance $\leq f(m,\varphi_2) - m$ from r. To ensure that φ_2 is still satisfied we make sure that the following stronger condition holds: every node b in the Σ-tree rooted at r is m'-GF^2-bisimilar to some b' in the original $\mathfrak{B}_r$ with distance $\leq f(m,\varphi_2) - m'$ from r. The construction is by a standard pumping argument. For $a,b \in B$ we say that a *blocks* b if $a \prec b$ and $(\mathfrak{B},a) \sim^{m',\text{succ}} (\mathfrak{B},b)$ and there is no $b' \prec b$ such that there is an a' with $a' \prec b'$ and $(\mathfrak{B},a') \sim^{m',\text{succ}} (\mathfrak{B},b')$. The universe B_r^u of $\mathfrak{B}_r^u$ is the set of words $a_0 \cdots a_n$ with $a_0,\ldots,a_n$ in $\mathfrak{B}_r^u$ and $a_0 = r$ such that either a_{i+1} is a successor of a_i or there is a successor b_{i+1} of a_i such that a_{i+1} blocks b_{i+1}. Let $\text{tail}(a_0 \ldots a_n) = a_n$. For every unary R and $w \in A_r^u$ we set $w \in R^{\mathfrak{A}_r^u}$ if $\text{tail}(w) \in R^{\mathfrak{A}}$ and for every binary R we set for $w \in A_r^u$: $(w,w) \in R^{\mathfrak{A}_r^u}$ if $(\text{tail}(w),\text{tail}(w)) \in R^{\mathfrak{A}}$ and for $wb \in A_r^u$:

- $(w,wb) \in R^{\mathfrak{A}_r^u}$ if $(\text{tail}(w),b) \in R^{\mathfrak{A}}$ or there is an a such that b blocks a and $(\text{tail}(w),a) \in R^{\mathfrak{A}}$;
- $(wb,w) \in R^{\mathfrak{A}_a^u}$ if $(b,\text{tail}(w)) \in R^{\mathfrak{A}}$ or there is an a such that b blocks a and $(a,\text{tail}(w)) \in R^{\mathfrak{A}}$.

We now replace $\mathfrak{B}_r$ by $\mathfrak{B}_r^u$ in $\mathfrak{B}$. In more detail, take the disjoint union $\mathfrak{B}^d$ of all $\mathfrak{B}_r^u$, r the root of a Σ-tree in $\mathfrak{B}$. Then add (recursively) witnesses for guarded existential quantifiers not involving binary predicates from Σ to $\mathfrak{B}^d$: take for any w in $\mathfrak{B}_r^u$ and any 1-type t for φ_2 that is realized in some node c in $\mathfrak{B}$ such that $\{R \mid (\text{tail}(w),c) \in R^{\mathfrak{B}}$ or $(c,\text{tail}(w)) \in R^{\mathfrak{B}}\}$ is non-empty and contains no predicate in Σ the root r' of a structure $\mathfrak{B}_{r'}^u$ such that r' realizes t in $\mathfrak{B}_{r'}^u$. Then add to $\mathfrak{B}^d$ a new copy of $\mathfrak{B}_{r'}^u$ and connect r' to b by adding for any binary predicate R the pair (r,r') to $R^{\mathfrak{B}_d}$ if $(\text{tail}(w),c) \in R^{\mathfrak{B}}$ and the pair (r',r) to $R^{\mathfrak{B}_d}$ if $(c,\text{tail}(w)) \in R^{\mathfrak{B}}$. We apply this procedure recursively to the new structure (in a fair way) and obtain the desired structure $\mathfrak{B}'$ as the limit of the resulting sequence of structures.

($\Rightarrow$) Assume that $\varphi_1 \models_\Sigma \varphi_2$. Let $\mathfrak{A}$ be a regular forest model $\mathfrak{A}$ of φ_1 that has finite outdegree and let $A_\perp \subseteq A$ be such that $A_\perp \cap \rho$ is infinite for any maximal infinite Σ-path ρ in $\mathfrak{A}$. By Theorem 26, there is a model $\mathfrak{B}$ of φ_2 such that

1. for every $a \in A$ there is a $b \in B$ such that $(\mathfrak{A},a) \sim_\Sigma (\mathfrak{B},b)$
2. for every $b \in B$, one of the following holds:
   a. there is an $a \in A$ such that $(\mathfrak{A},a) \sim_\Sigma (\mathfrak{B},b)$;

b. there are $a_\perp, a_0, a_1, \ldots, a'_0, a'_1, \cdots \in A$ such that $a_\perp \prec a_0 \prec a_1 \prec \cdots$ and, for all $i \geq 0$, $a_i \prec a'_i$ and $(\mathfrak{A}, a'_i) \sim_\Sigma^{a_\perp} (\mathfrak{B}, b)$.

Let t be a 1-type for φ_2 realized by some $b \in B$. We have to find an $a \in A$ such that $(\mathfrak{A}, a) \sim_\Sigma^{A_\perp} (\mathfrak{B}, b)$. If there is an $a \in A$ such that $(\mathfrak{A}, a) \sim_\Sigma (\mathfrak{B}, b)$ then we are done as $(\mathfrak{A}, a) \sim_\Sigma^{A_\perp} (\mathfrak{B}, b)$ follows. Otherwise there are $a_\perp, a_0, a_1, \ldots, a'_0, a'_1, \cdots \in A$ such that $a_\perp \prec a_0 \prec a_1 \prec \cdots$ and, for all $i \geq 0$, $a_i \prec a'_i$ and $(\mathfrak{A}, a'_i) \sim_\Sigma^{a_\perp} (\mathfrak{B}, b)$. Then let ρ be a Σ-path containing $a_\perp, a_0, a_1, \ldots$. $A_\perp \cap \rho$ is infinite and so we can choose an a'_i such that there are at least two elements of $A_\perp$ on the path from $a_\perp$ to a'_i. It follows from the definition of $\sim_\Sigma^{A_\perp}$ that $(\mathfrak{A}, a'_i) \sim_\Sigma^{A_\perp} (\mathfrak{B}, b)$, as required. ❑

D Proofs for Section 5

We construct the required 2ATAs.

▶ **Lemma 27.** *Let φ_1 be a GF^2-sentence. There is a 2ATA $\mathcal{A}_1$ that accepts a Θ-labeled tree (T, L) iff $\mathfrak{A}_{(T,L)}$ is a model of φ_1.*

We assume that in all subformulas of φ_1 of the form $\exists \mathbf{y}(\alpha(\mathbf{x}, \mathbf{y}) \wedge \varphi(\mathbf{x}, \mathbf{y}))$ and $\forall \mathbf{y}(\alpha(\mathbf{x}, \mathbf{y}) \rightarrow \varphi(\mathbf{x}, \mathbf{y}))$, $\mathbf{y}$ consists of exactly one variable and $\alpha(\mathbf{x}, \mathbf{y})$ is a relational atom with two variables or an equality atom. This can be done w.l.o.g. because each sentence $\exists xy\varphi(x, y)$ can be rewritten into $\exists x(x = x \wedge \exists y \varphi(x, y))$, each sentence $\exists x(\alpha(x) \wedge \varphi(x))$ with α a relational atom can be rewritten into $\exists x(x = x \wedge \alpha(x) \wedge \varphi(x))$, and likewise for universal quantifiers. We further assume that φ_1 has no subformulas of the form $\exists x(x = y \wedge \varphi(x, y))$ with $x \neq y$; such formulas are equivalent to $\varphi[y/x]$, that is, the result of replacing in φ all occurrences of x with y. The result of these assumptions is that each formula $\exists \mathbf{y}(\alpha(\mathbf{x}, \mathbf{y}) \wedge \varphi(\mathbf{x}, \mathbf{y}))$ takes the form $\exists x(x = x \wedge \psi(x))$ or $\exists x \psi(x, y)$, and likewise for universally quantified formulas. We define $\mathcal{A}_1 = (Q_1, \Theta, q_{\varphi_1}, \delta_1, \Omega_1)$ where

$$Q_1 = \{q_{\varphi(x)} \mid \varphi(x) \in \mathsf{cl}(\varphi_1)\} \cup \{q_{\varphi(x,\underline{y})}, q_{\varphi(\underline{x},y)} \mid \varphi(x, y) \in \mathsf{cl}(\varphi_1)\}$$

and Ω_1 assigns two to all states except those of the form $q_{\exists x(x=x \wedge \psi(x))}$, to which it assigns one. The underlining in states of the form $q_{\varphi(x,\underline{y})}$ and $q_{\varphi(\underline{x},y)}$ serves as a marking of the variable that is bound to the tree node to which the state is assigned. We define the transition function δ_1 as follows, for each $\sigma = (\tau, M)$:

$$\delta_1(q_{Az}, \sigma) = \begin{cases} \mathsf{true} & \text{if } Ay \in \tau \\ \mathsf{false} & \text{otherwise} \end{cases}$$

$$\delta_1(q_{\neg Az}, \sigma) = \begin{cases} \mathsf{true} & \text{if } Ay \notin \tau \\ \mathsf{false} & \text{otherwise} \end{cases}$$

$$\begin{aligned}
\delta_1(q_{\varphi(z) \circ \psi(z)}, \sigma) &= q_{\varphi(z)} \circ q_{\psi(z)} \\
\delta_1(q_{\exists z(z=z \wedge \psi(z))}) &= q_{\psi(z)} \vee \langle -1 \rangle q_{\exists z(z=z \wedge \psi(z))} \vee \Diamond q_{\exists z(z=z \wedge \psi(z))} \\
\delta_1(q_{\forall z(z=z \rightarrow \psi(z))}) &= q_{\psi(z)} \wedge [-] q_{\forall z(z=z \rightarrow \psi(z))} \wedge \Box q_{\forall z(z=z \rightarrow \psi(z))} \\
\delta_1(q_{\exists z' \varphi(z,z')}, \sigma) &= \Diamond q_{\varphi(z,\underline{z'})} \vee q_{\varphi(z',\underline{z})} \\
\delta_1(q_{\forall z' \varphi(z,z')}, \sigma) &= \Box q_{\varphi(z,\underline{z'})} \wedge q_{\varphi(z',\underline{z})}
\end{aligned}$$

$$\delta_1(q_{Rz\underline{z}'}, \sigma)) = \begin{cases} \mathsf{true} & \text{if } Rxy \in \tau \\ \mathsf{false} & \text{otherwise} \end{cases}$$

$$\delta_1(q_{R\underline{z}z'}, \sigma)) = \begin{cases} \mathsf{true} & \text{if } Ryx \in \tau \\ \mathsf{false} & \text{otherwise} \end{cases}$$

$$\delta_1(q_{\neg Rz\underline{z}'}, \sigma)) = \begin{cases} \mathsf{true} & \text{if } Rxy \notin \tau \\ \mathsf{false} & \text{otherwise} \end{cases}$$

$$\delta_1(q_{\neg R\underline{z}z'}, \sigma)) = \begin{cases} \mathsf{true} & \text{if } Ryx \notin \tau \\ \mathsf{false} & \text{otherwise} \end{cases}$$

$$\begin{aligned} \delta_1(q_{\varphi(z,\underline{z}')\circ\psi(z,\underline{z}')}, \sigma) &= q_{\varphi(z,\underline{z}')} \circ q_{\psi(z,\underline{z}')} \\ \delta_1(q_{\varphi(z,\underline{z}')\circ\psi(z)}, \sigma) &= q_{\varphi(z,\underline{z}')} \circ \langle -1\rangle q_{\psi(z)} \\ \delta_1(q_{\varphi(z,\underline{z}')\circ\psi(z')}, \sigma) &= q_{\varphi(z,\underline{z}')} \circ q_{\psi(z')} \\ \delta_1(q_{\varphi(z)\circ\psi(z')}, \sigma) &= \langle -1\rangle q_{\varphi(z)} \circ q_{\psi(z')} \end{aligned}$$

where σ ranges over Θ, z, z' range over $\{x, y\}$, and $\circ$ ranges over $\{\wedge, \vee\}$. With $\varphi(z', z)$, we mean the result of exchanging in $\varphi(z, z')$ the variables z and z', and $\overline{\varphi(z, z')}$ denotes the negation normal form of the negation of $\varphi(z, z')$.

We now complete the construction of the 2ATA $\mathcal{A}_2$. It remains to implement the obligation represented by states of the form t^0, that is, the existence of $A_\perp$-delimited $\mathrm{GF}^2(\Sigma)$-bisimulations. Recall that such a bisimulation consists of two relations $\sim_\Sigma^{A_\perp,0}$ and $\sim_\Sigma^{A_\perp,1}$, each of which behaves essentially like a $\mathrm{GF}^2(\Sigma)$-bisimulation except in some special cases that pertain to the $A_\perp$-marking of one of the involved structures, which in this case is the structure $\mathfrak{A}_{(T,L)}$. To deal with $\sim_\Sigma^{A_\perp,0}$ and $\sim_\Sigma^{A_\perp,1}$, we take copies q^0 and q^1 of every state q that is of the form t, $t_\downarrow$, $t_\uparrow$, λ, and $\lambda_\uparrow$, and also copies of the above block of transitions, modified in a suitable way to take care of the special cases. This is implemented for $\sim_\Sigma^{A_\perp,0}$ by the following transitions:

$$\begin{array}{rcll}
\delta_2(t^0, (\tau, M)) & = & t^0_\uparrow \wedge \Box t^0_\downarrow \wedge \bigvee\limits_{T \mid t \approx T} \bigwedge\limits_{\lambda \in T} (\Diamond \lambda^0 \vee \lambda^0_\uparrow) & \text{if } \tau_y =_\Sigma t \\
\delta_2(t^0, (\tau, M)) & = & \mathsf{false} & \text{if } \tau_y \neq_\Sigma t \\
\delta_2(t^0_\downarrow, (\tau, M)) & = & \mathsf{true} & \text{if } \tau \text{ is not } \Sigma\text{-guarded} \\
\delta_2(t^0_\downarrow, (\tau, M)) & = & \bigvee\limits_{\lambda \mid t \approx \lambda \wedge \tau =_\Sigma \lambda} \lambda^0_y & \text{if } \tau \text{ is } \Sigma\text{-guarded} \\
\delta_2(t^0_\uparrow, (\tau, M)) & = & \mathsf{true} & \text{if } \tau \text{ is not } \Sigma\text{-guarded} \\
\delta_2(t^0_\uparrow, (\tau, M)) & = & \bigvee\limits_{\lambda \mid t \approx \lambda \wedge \tau =_\Sigma \lambda^-} [-1]\lambda^M_y & \text{if } \tau \text{ is } \Sigma\text{-guarded} \\
\delta_2(\lambda^0, (\tau, M)) & = & \lambda^0_y & \text{if } \lambda \text{ is } \Sigma\text{-guarded and } \tau =_\Sigma \lambda \\
\delta_2(\lambda^0, (\tau, M)) & = & \mathsf{false} & \text{if } \lambda \text{ is } \Sigma\text{-guarded and } \tau \neq_\Sigma \lambda \\
\delta_2(\lambda^0, (\tau, M)) & = & \lambda^?_y & \text{if } \lambda \text{ is not } \Sigma\text{-guarded} \\
\delta_2(\lambda^0_\uparrow, (\tau, M)) & = & \langle -1\rangle(\lambda_y)_\& & \text{if } \lambda \text{ is } \Sigma\text{-guarded and } \tau =_\Sigma \lambda^- \\
\delta_2(\lambda^0_\uparrow, (\tau, M)) & = & \mathsf{false} & \text{if } \lambda \text{ is } \Sigma\text{-guarded and } \tau \neq_\Sigma \lambda^- \\
\delta_2(\lambda^0_\uparrow, (\tau, M)) & = & \lambda^?_y & \text{if } \lambda \text{ is not } \Sigma\text{-guarded} \\
\delta_2(t_\&, (\tau, M)) & = & t^M &
\end{array}$$

The transitions for $\sim_\Sigma^{A_\perp,1}$ are as follows:

$$
\begin{array}{rcll}
\delta_2(t^1,(\tau,1)) & = & \big(t^? \wedge t^1_\uparrow \wedge \Box t^1_\downarrow \wedge \bigvee_{T\mid t\approx T} \bigwedge_{\lambda\in T}(\Diamond\lambda^1 \vee \lambda^1_\uparrow)\big) & \text{if } \tau_y =_\Sigma t \\
 & & \vee\ (t^? \wedge \langle -1\rangle q_\perp) & \\
\delta_2(t^1,(\tau,0)) & = & \big(t^? \wedge t^1_\uparrow \wedge \Box t^0_\downarrow \wedge \bigvee_{T\mid t\approx T} \bigwedge_{\lambda\in T}(\Diamond\lambda^0 \vee \lambda^1_\uparrow)\big) & \text{if } \tau_y =_\Sigma t \\
 & & \vee\ (t^? \wedge \langle -1\rangle q_\perp) & \\
\delta_2(t^1,(\tau,M)) & = & \langle -1\rangle q_\perp & \text{if } \tau_y \neq_\Sigma t \\
\delta_2(q_\perp,(\tau,0)) & = & \mathsf{false} & \\
\delta_2(q_\perp,(\tau,1)) & = & \mathsf{true} & \\
\delta_2(t^1_\downarrow,(\tau,M)) & = & \mathsf{true} & \text{if } \tau \text{ is not } \Sigma\text{-guarded} \\
\delta_2(t^1_\downarrow,(\tau,M)) & = & \bigvee_{\lambda\mid t\approx\lambda\wedge\tau=_\Sigma\lambda} \lambda^1_y & \text{if } \tau \text{ is } \Sigma\text{-guarded} \\
\delta_2(t^1_\uparrow,(\tau,M)) & = & \mathsf{true} & \text{if } \tau \text{ is not } \Sigma\text{-guarded} \\
\delta_2(t^1_\uparrow,(\tau,M)) & = & \bigvee_{\lambda\mid t\approx\lambda\wedge\tau=_\Sigma\lambda^-} [-1]\lambda^1_y & \text{if } \tau \text{ is } \Sigma\text{-guarded} \\
\delta_2(\lambda^1,(\tau,M)) & = & \lambda^1_y & \text{if } \lambda \text{ is } \Sigma\text{-guarded and } \tau =_\Sigma \lambda \\
\delta_2(\lambda^1,(\tau,M)) & = & \mathsf{false} & \text{if } \lambda \text{ is } \Sigma\text{-guarded and } \tau \neq_\Sigma \lambda \\
\delta_2(\lambda^1,(\tau,M)) & = & \lambda^?_y & \text{if } \lambda \text{ is not } \Sigma\text{-guarded} \\
\delta_2(\lambda^1_\uparrow,(\tau,M)) & = & \langle -1\rangle\lambda^1_y & \text{if } \lambda \text{ is } \Sigma\text{-guarded and } \tau =_\Sigma \lambda^- \\
\delta_2(\lambda^1_\uparrow,(\tau,M)) & = & \mathsf{false} & \text{if } \lambda \text{ is } \Sigma\text{-guarded and } \tau \neq_\Sigma \lambda^- \\
\delta_2(\lambda^1_\uparrow,(\tau,M)) & = & \lambda^?_y & \text{if } \lambda \text{ is not } \Sigma\text{-guarded}
\end{array}
$$

▶ **Lemma 28.** *$\mathcal{A}_2$ satisfies the condition from Lemma 13.*

Proof. "⇐". Let (T,L) be a Θ-labeled tree and let $\mathfrak{B}$ be a model of φ_2 such that Conditions 1 and 2 of Theorem 12 are satisfied when $\mathfrak{A}$ is replaced with $\mathfrak{A}_{(T,L)}$ (and when $A_\perp$ is the set described by the second component of the L-labels). We argue that $\mathfrak{B}$ can be used to guide a run of $\mathcal{A}_2$ on (T,L) so that it is accepting.

In this run, $\mathcal{A}_2$ starts with choosing the 0-type s realized by $\mathfrak{B}$. Then, for each $\exists x\varphi(x) \in s$, we guide $\mathcal{A}_2$ to proceed in state $t^?$, where t is the 1-type of some element $b \in B$ with $\mathfrak{B} \models \varphi(b)$. By Condition 2 of Theorem 12, there is a $w \in A_{(T,L)}$ such that $\mathsf{tp}_{\mathfrak{B}}(b) = t$ and $(\mathfrak{A}_{(T,L)}, w) \sim^{A_\perp} (\mathfrak{B}, b)$. In the search state $t^?$, we guide the run to reach w and switch to state t^0 there. The automaton also sends a copy in state s to each node $w \in A_{(T,L)}$. By Condition 1 of Theorem 12, there is a $b \in B$ such that $(\mathfrak{A}_{(T,L)}, w) \sim_\Sigma (\mathfrak{B}, b)$. We guide the run to proceed in state t, the 1-type of b.

At this point, the automaton needs to satisfy two kinds of obligations:

1. states t true at a node $w \in A_{(T,L)}$ representing the obligation to verify that there is a $b \in B$ with 1-type t and such that $(\mathfrak{A}_{(T,L)}, w) \sim_\Sigma (\mathfrak{B}, b)$ and
2. states t^0 true at a node $w \in A_{(T,L)}$ representing the obligation to verify that there is a $b \in B$ with 1-type t and such that $(\mathfrak{A}_{(T,L)}, w) \sim_\Sigma^{A_\perp} (\mathfrak{B}, b)$.

Note that we have guided the run so that the required bisimulations indeed exist and therefore we can use them to further guide the run. We only consider Case 1 above, thus concentrating on states of the form t, $t_\downarrow$, $t_\uparrow$, λ, and $\lambda_\uparrow$. Suppose the automaton is in state t at node w. By the way in which we guide the run, there is then a $b \in B$ with 1-type t and such that $(\mathfrak{A}_{(T,L)}, w) \sim_\Sigma (\mathfrak{B}, b)$. We guide the run to select as T the set of all guarded 2-types λ such that $\mathfrak{B} \models (\exists y\lambda(x,y))(b)$. For each such λ, there must be a $b' \in B$ and a $v \in A_{(T,L)}$ with

$\mathfrak{B} \models \lambda(b, b')$ and $(\mathfrak{A}_{(T,L)}, v) \sim_\Sigma (\mathfrak{B}, b')$ where v is either the predecessor of w or a successor of it. In the former case, we guide the automaton to switch to state $\lambda_\uparrow$ and in the latter, we guide it to execute $\Diamond\lambda$. When the automaton was sent in state $t_\downarrow$ to a successor v of w, then there must be a $b' \in B$ such that $(\mathfrak{A}_{(T,L)}, v) \sim_\Sigma (\mathfrak{B}, b')$ and $\mathfrak{B} \models \lambda(b, b')$ for some guarded 2-type λ. Guide the run to choose λ. The decision to be taken for states $t_\uparrow$ is handled very similarly.

"$\Rightarrow$". Let (T, L) be a Θ-labeled tree that is accepted by $\mathcal{A}_2$. Then there is an accepting run (T_r, r) of $\mathcal{A}_2$ on (T, L). We show how to use (T_r, r) to construct a model $\mathfrak{B}$ of φ_2 such that Conditions 1 and 2 of Theorem 12 are satisfied when $\mathfrak{A}$ is replaced with $\mathfrak{A}_{(T,L)}$. Along with $\mathfrak{B}$, we construct the following objects:

- a $\mathrm{GF}^2(\Sigma)$-bisimulation $\sim$ between $\mathfrak{A}_{(T,L)}$ and $\mathfrak{B}$ which witnesses that Condition 1 of Theorem 12 is satisfied,
- two relations $\sim^{A_\perp,0}$ and $\sim^{A_\perp,1}$ that form an $A_\perp$-delimited $\mathrm{GF}^2(\Sigma)$-bisimulation between $\mathfrak{A}_{(T,L)}$ and $\mathfrak{B}$, where $A_\perp \subseteq \mathfrak{A}_{(T,L)}$ is the subset defined by the second component of L, and which witness that Condition 2 of Theorem 12 is satisfied, and
- a function μ that assigns to each element of $\mathfrak{B}$ the 1-type that we aim to realize there.

Throughout the construction, we make sure that the following invariants are satisfied:

1. if $(w, b) \in \sim$, then the label $(w, \mu(b))$ occurs in (T_r, r);
2. if $(w, b) \in \sim^{A_\perp,i}$, $i \in \{0, 1\}$, then the label $(w, \mu(b)^0)$ occurs in (T_r, r).

The start of the construction is as follows:

- for each label (w, t) that occurs in (T_r, r), introduce an element b of B, add (w, b) to $\sim$, and set $\mu(b) = t$;
- for each label (w, t^0) that occurs in (T_r, r), introduce an element b of B, add (w, b) to $\sim^{A_\perp,0}$, and set $\mu(b) = t$.

We then iteratively extend $\mathfrak{B}$, $\sim$, $\sim^{A_\perp,0}$, $\sim^{A_\perp,1}$, and μ, obtaining the desired structure and bisimulations in the limit. In each step, process every $b \in B$ that has not been processed in a previous round. There are three cases.

Case (a). There is a $(w, b) \in \sim$. By Invariant 1, we find a node $x \in T_r$ such that $r(x) = (w, \mu(b))$. We perform two steps:

- For every predecessor or successor v of w in T with $\mathrm{at}^\Sigma_{\mathfrak{A}_{(T,L)}}(w, v)$ guarded, there must be a 2-type λ such that $\mu(b) \approx \lambda$, (v, λ_y) occurs as a label in (T_r, r), and $\mathrm{at}^\Sigma_{\mathfrak{A}_{(T,L)}}(w, v) =_\Sigma \lambda$. Extend $\mathfrak{B}$ with a new element b', extend the interpretation of the predicates in $\mathfrak{B}$ such that $\mathrm{at}^\Sigma_{\mathfrak{B}}(w, v) =_\Sigma \lambda$, set $\mu(b') = \lambda_y$, and extend $\sim$ with (v, b').
- There must be a set T of guarded 2-types such that $t \approx T$ and for every $\lambda \in T$, there is a predecessor or successor v of w in T such that $\mu(b) \approx \lambda$, (v, λ_y) occurs as a label in (T_r, r), and $\mathrm{at}^\Sigma_{\mathfrak{A}_{(T,L)}}(w, v) =_\Sigma \lambda$. Extend $\mathfrak{B}$ with a new element b' (for every λ), extend the interpretation of the predicates in $\mathfrak{B}$ such that $\mathrm{at}^\Sigma_{\mathfrak{B}}(w, v) =_\Sigma \lambda$, set $\mu(b') = \lambda_y$, and extend $\sim$ with (v, b').

Case (b). There is a $(w, b) \in \sim^{A_\perp,0}$. By Invariant 2, we find a node $x \in T_r$ such that $r(x) = (w, \mu(b)^0)$. We can now proceed exactly as in Case (a) except that, in both subcases, we add (v, b') to $\sim^{A_\perp,1}$ if v is a predecessor of w and $v \in A_\perp$, and to $\sim^{A_\perp,0}$ otherwise.

Case (c). There is a $(w, b) \in \sim^{A_\perp,1}$. By Invariant 2, we find a node $x \in T_r$ such that $r(x) = (w, \mu(b)^1)$. If the predecessor of w is not in $A_\perp$, then we again proceed as in Case (a) except that, in both subcases, we add (v, b') to $\sim^{A_\perp,0}$ if v is a sucessor of w and $w \in A_\perp$, and to $\sim^{A_\perp,1}$ otherwise. If the predecessor of w is in $A_\perp$, then we we also proceed as in Case (a), but do not add (v, b') to any of the constructed bisimulations.

Case (d). None of the above cases applies. Then we proceed as in Case (a), again not adding (v, b') to any of the constructed bisimulations.

It can be verified that, as intended the structure $\mathfrak{B}$ obtained in the limit is a model of φ_2, that the relation $\sim$ is a $\mathrm{GF}^2(\Sigma)$-bisimulation, and that $\sim^{A_\perp,0}, \sim^{A_\perp,1}$ form an $A_\perp$-delimited $\mathrm{GF}^2(\Sigma)$-bisimulation. ❑

Recall that we define the overall 2ATA $\mathcal{A}$ so that it accepts $L(\mathcal{A}_1) \cap \overline{L(\mathcal{A}_2)} \cap L(\mathcal{A}_3)$. Using Theorem 12, it can be verified that, as intended, $\varphi_1 \models_{\mathrm{GF}^2(\Sigma)} \varphi_2$ iff $L(\mathcal{A}) = \emptyset$. Note that for the "only if" direction, we have to show that $L(\mathcal{A}) \neq \emptyset$ implies that there is a *regular* forest model of φ_1 that satisfies the negation of the conditions in Theorem 12. As is the case for other kinds of tree automata, also for the 2ATA $\mathcal{A}$ it can be shown that $L(\mathcal{A}) \neq \emptyset$ implies that $\mathcal{A}$ accepts a regular Θ-labeled tree (T, L). The corresponding structure $\mathfrak{A}_{(T,L)}$ must then also be regular.

We show that Σ-entailment, Σ-inseparability, and conservative extensions in GF^2 are 2ExpTime-hard. The proof is by reduction of the word problem for exponentially space bounded alternating Turing machines (ATMs). An *ATM* is of the form $M = (Q, \Theta, \Gamma, q_0, \Delta)$. The set of *states* $Q = Q_\exists \uplus Q_\forall \uplus \{q_a\} \uplus \{q_r\}$ consists of *existential states* from $Q_\exists$, *universal states* from $Q_\forall$, an *accepting state* q_a, and a *rejecting state* q_r; Θ is the *input alphabet* and $\Gamma \supset \Theta$ the *work alphabet* that contains a *blank symbol* $\Box \notin \Theta$; $q_0 \in Q_\exists$ is the *starting* state; and the *transition relation* Δ is of the form $\Delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{L, R\}$. We write $\Delta(q, a)$ for $\{(q', b, M) \mid (q, a, q', b, M) \in \Delta\}$ and assume that $\Delta(q, b) = \emptyset$ for all $q \in \{q_a, q_r\}$ and $b \in \Gamma$.

A *configuration* of an ATM is a word wqw' with $w, w' \in \Gamma^*$ and $q \in Q$. The intended meaning is that the one-side infinite tape contains the word ww' with only blanks behind it, the machine is in state q, and the head is on the symbol just after w. The *successor configurations* of a configuration wqw' are defined in the usual way in terms of the transition relation Δ. A *halting configuration* (resp. *accepting configuration*) is of the form wqw' with $q \in \{q_a, q_r\}$ (resp. $q = q_a$).

A *computation tree* of an ATM M on input w is a tree whose nodes are labeled with configurations of M on w, such that the descendants of any non-leaf labeled by a universal (resp. existential) configuration include all (resp. one) of the successor configurations of that configuration. A computation tree is *accepting* if the root is labeled with the *initial configuration* $q_0 w$ for w and all leaves with accepting configurations. An ATM M accepts input w if there is a computation tree of M on w.

Take an exponentially space bounded ATM M whose word problem is 2ExpTime-hard [8]. We may w.l.o.g. assume that the length of every computation path of M on $w \in \Theta^n$ is bounded by 2^{2^n}. We can also assume that for each $q \in Q_\forall \cup Q_\exists$ and each $a \in \Gamma$, the set $\Delta(q, a)$ has exactly two elements. We assume that these elements are ordered, i.e., $\Delta(q, a)$ is an ordered pair $((q_L, b_L, M_L), (q_R, b_R, M_R))$. Furthermore, we assume that M never attempts to move left on the left-most tape cell.

Let $w = a_0 \cdots a_{n-1} \in \Theta^*$ be an input to M. In the following, we construct GF^2 sentences φ_1 and φ_2 such that $\varphi_1 \wedge \varphi_2$ is a conservative extension of φ_1 if and only if M does not accept w. Informally, the main idea is to construct φ_1 and φ_2 such that models of sentences that witness non-conservativity describe an accepting computation tree of M on w. In such models, each domain element represents a tape cell of a configuration of M, the binary predicate N indicates moving to the next tape cell in the same configuration, and the binary predicates L and R indicate moving to left and right successor configurations in accepting configuration trees. Thus, each node of the computation tree (that is, each configuration) is spread out over a sequence of nodes in the model. We actually assume that every non-halting configuration has two successor configurations, also when its state is existential. This can of course easily be achieved by duplicating subtrees in computation trees. The following predicates are used in φ_1:

- a unary predicate P to mark the root of computation trees;
- binary predicates N, R, L, as explained above;
- unary predicates $C_0, \ldots, C_{n-1}$ that represent the bits of a binary counter which identifies tape positions;
- a unary predicate F that marks the topmost configuration in the configuration tree;
- unary predicates S_a, $a \in \Gamma$, to represent the tape content of cells that are not under the head;
- unary predicates $S_{q,a}$, $q \in Q$ and $a \in \Gamma$, to represent the state of a configuration, the head position, and the tape content of the cell that is under the head;
- unary predicates S^p_a and $S^p_{q,a}$, with the ranges of q and a as above, to represent the same information, but for the previous configuration in the tree instead of for the current one;
- unary predicates $Y_{L,q,a,M}$ and $Y_{R,q,a,M}$, $q \in Q$, $a \in \Gamma$, $M \in \{L, R\}$, to record the transition to be executed in the subsequent configurations;
- unary predicates $Y_{q,a,M}$, with the ranges of q, a, M as above, to record the transition executed to reach the current configuration.

The sentence φ_2 uses some additional unary predicates, including $C'_0, \ldots, C'_{n-1}$ to implement another counter whose purpose is explained below.

The sentences φ_1 and φ_2 are shown in Figures 1 and 2, respectively, where q and q' range over Q, a, b, b' over Γ, M and P over $\{L, R\}$, T, T_L, T_R over $Q \times \Gamma \times \{L, R\}$, and α over $\Gamma \cup (Q \times \Gamma)$. The formula $\varphi_{C=i}(x)$, which is easily worked out in detail, expresses that the value of the binary counter implemented by $C_0, \ldots, C_{n-1}$ has value exactly i at x, and likewise for $\varphi_{C<i}(x)$ and $\varphi_{C\geq i}(x)$, and for the primed versions in φ_2 which refer to the counter implemented by $C'_0, \ldots, C'_{n-1}$. The formula $\varphi_{C++}(x, y)$ expresses that the counter value at y is obtained from the counter value at x by incrementation modulo 2^n. Again, we omit the details.

Let us walk through φ_1 and φ_2 and give some intuition of what the various conjuncts are good for. In φ_1, Lines (1) to (4) ensure that at an element that satisfies P, there is an infinite tree of the expected pattern: first $2^n - 1$ N-edges without branching, then a binary branching of an L-edge and an R-edge, then $2^n - 1$ N-edges without branching, and so on, ad infinitum. Of course, a computation tree will be represented using only a finite initial piece of this infinite tree. These conjuncts also set up the counter C so that it identifies the position of tape cells and the marker F so that it identifies the topmost configuration in the tree. Line (5) says that every cell is labeled with exactly one symbol and that the state is unique (locally to one cell; there is no need to express the same globally for the entire configuration), and Line (6) says the same for the representation of the previous configuration. Lines (7) to (9) make sure that the topmost configuration in the infinite tree is the initial configuration of M on input w. Lines (10) and (11) choose transitions to execute and Lines (12) to (14) propagate this choice down to the subsequent configurations. Assume that the predicates of the S^p_a and $S^p_{q,a}$ indeed represent the previous configuration, Lines (15) to (20) then implement the chosen transitions. Line (21) says that we do never see a rejecting halting configuration.

Now for φ_2. Essentially, we want to achieve that a sentence is a witness for non-conservativity if and only if it expresses that its models contain (a representation of) an accepting computation tree of M on w whose root is labeled with P. This is achieved by

$$\forall x(Px \to \varphi_{C=0}(x)) \tag{1}$$
$$\forall x\big(\varphi_{C<2^n-1}(x) \to \big(\exists y Nxy \wedge \forall y(Nxy \to \varphi_{C++}(x,y))\big)\big) \tag{2}$$
$$\forall x\big(\varphi_{C=2^n-1}(x) \to (\exists y Lxy \wedge \forall y(Lxy \to \varphi_{C=0}(y)) \wedge \exists y Rxy \wedge \forall y(Rxy \to \varphi_{C=0}(y)))\big) \tag{3}$$
$$\forall x\big((Px \to Fx) \wedge \forall y(Nxy \to Fy)\big) \tag{4}$$
$$\forall x \bigvee_{\alpha \in \Gamma \cup (Q \times \Gamma)} \big(S_\alpha x \wedge \bigwedge_{\beta \in (\Gamma \cup (Q \times \Gamma)) \setminus \{\alpha\}} \neg S_\beta x\big) \tag{5}$$
$$\forall x \bigvee_{\alpha \in \Gamma \cup (Q \times \Gamma)} \big(S^p_\alpha x \wedge \bigwedge_{\beta \in (\Gamma \cup (Q \times \Gamma)) \setminus \{\alpha\}} \neg S^p_\beta x\big) \tag{6}$$
$$\forall x\big((Fx \wedge \varphi_{C=0}(x)) \to S_{q_0,a_0}x\big) \tag{7}$$
$$\forall x\big((Fx \wedge \varphi_{C=i}(x)) \to S_{a_i}x\big) \qquad \text{for } 1 \le i < n \tag{8}$$
$$\forall x\big((Fx \wedge \varphi_{C\ge n}(x)) \to S_{\Box}x\big) \tag{9}$$
$$\forall x\big(S_{q,a}x \to (Y_{L,T_L}x \wedge Y_{R,T_R}x)\big) \qquad \text{if } \Delta(q,a) = (T_L, T_R),\ q \in Q_\forall \tag{10}$$
$$\forall x\big(S_{q,a}x \to ((Y_{L,T_L}x \wedge Y_{R,T_L}x) \vee (Y_{L,T_R}x \wedge Y_{R,T_R}x))\big) \qquad \text{if } \Delta(q,a) = (T_L, T_R),\ q \in Q_\exists \tag{11}$$
$$\forall x\big(Y_{P,T}x \to \forall y(Nxy \to Y_{P,T}y)\big) \tag{12}$$
$$\forall x\big(Y_{P,T}x \to \forall y(Pxy \to Y_T y)\big) \tag{13}$$
$$\forall x\big(Y_T x \to \forall y(Nxy \to Y_T y)\big) \tag{14}$$
$$\forall x\big((Y_{q,a,M}x \wedge S^p_{q',b}x) \to S_a x\big) \tag{15}$$
$$\forall x\big((Y_{q,a,L}x \wedge S^p_b x \wedge \exists y(Nxy \wedge S^p_{q',a'}y)) \to S_{q,b}x\big) \tag{16}$$
$$\forall x\big((Y_{q,a,R}x \wedge S^p_b x \wedge \exists y(Nxy \wedge S^p_{q',a'}y)) \to S_b x\big) \tag{17}$$
$$\forall x\big((Y_{q,a,R}x \wedge S^p_b x \wedge \exists y(Nyx \wedge S^p_{q',a'}y)) \to S_{q,b}x\big) \tag{18}$$
$$\forall x\big((Y_{q,a,L}x \wedge S^p_b x \wedge \exists y(Nyx \wedge S^p_{q',a'}y)) \to S_b x\big) \tag{19}$$
$$\forall x\big((\exists y(Nxy \wedge S^p_b y)) \wedge S^p_a x \wedge \exists y(Nyx \wedge S^p_{b'}y)) \to S_a x\big) \tag{20}$$
$$\forall x \neg S_{q_r,a}x \tag{21}$$

Figure 1 The conjuncts of the sentence φ_1.

designing φ_2 so that, whenever a tree model of φ_1 contains only instances of P that are not the root of such a computation tree, then this model can be extended to a model of φ_2 by assigning an interpretation to the additional predicates in φ_2. Note that φ_1 already enforces that, below any instance of P, there is a tree that satisfies almost all of the required conditions of an accepting computation tree. In fact, the only way in which that tree cannot be an accepting configuration tree is that the predicates S^p_a and $S^p_{q,a}$ do not behave in the expected way, that is, there is a configuration and a cell in this configuration that is labeled with S_α, $\alpha \in \Gamma \cup (Q \times \Gamma)$, and in one of the two subsequent configurations the same cell is not labeled with S^p_α. We thus design φ_2 so that it can be made true whenever the model contains such a defect. In Line (22), we select the place where the defect is. Line (23) ensures

$$\exists x Px \rightarrow \exists x Dx \tag{22}$$

$$\forall x\big(Dx \rightarrow (Mx \wedge \varphi_{C'=0}(x))\big) \tag{23}$$

$$\forall x\big((Dx \wedge S_\alpha x) \rightarrow Z_\alpha x\big) \tag{24}$$

$$\forall x\big((Mx \wedge \varphi_{C<2^n-1}(x) \wedge \varphi_{C'<2^n-1}(x) \wedge Z_\alpha x) \rightarrow \exists y(Nxy \wedge My \wedge Z_\alpha y \wedge \varphi_{C'++}(x,y))\big) \tag{25}$$

$$\forall x\big((Mx \wedge \varphi_{C=2^n-1}(x) \wedge \varphi_{C'<2^n-1}(x) \wedge Z_\alpha x) \rightarrow \tag{26}$$
$$\exists y(Lxy \wedge My \wedge Z_\alpha y \wedge \varphi_{C'++}(x,y)) \vee \exists y(Rxy \wedge My \wedge Z_\alpha y \wedge \varphi_{C'++}(x,y))\big)$$

$$\forall x\big((Mx \wedge \varphi_{C'=2^n-1}(x) \wedge Z_\alpha x) \rightarrow \exists y(Nxy \wedge \neg S^p_\alpha x)\big) \tag{27}$$

Figure 2 The conjuncts of the sentence φ_2.

that the counter C' starts counting with value zero at that place, and that the marker M is set there, too. Line (24) memorizes the content α of the cell in the upper configuration involved in the defect. Lines (25) to (27) propagate downwards the memorized content and make sure that, at the corresponding cell of at least one subsequent configuration (which is identified using the counter C'), we do not find S^p_α.

▶ **Lemma 29.**

1. *If M accepts w, then $\varphi_1 \wedge \varphi_2$ is not a GF^2-conservative extension of φ_1.*
2. *If there exists a $\mathsf{sig}(\varphi_1)$-structure that satisfies φ_1 and cannot be extended to a model of $\varphi_1 \wedge \varphi_2$, then M does not accept w.*

Proof.(sketch) For Point 1 assume that M accept w. Then there is an accepting computation tree of M on w. Let $\Sigma = \mathsf{sig}(\varphi_1)$. We can find a GF$^2(\Sigma)$-sentence ψ_1 which expresses that the model contains a (homomorphic image of a) finite tree which represents this configuration tree and whose root is labeled with P. We can also find a GF$^2(\Sigma)$-sentence ψ_2 which expresses that nowhere in the model there is a defect situation. It can be verified that $\psi_1 \wedge \psi_2$ is satisfiable w.r.t. φ_1, but not w.r.t. φ_2 because φ_2 requires the existence of a defect situation whenever the extension of P is non-empty.

For Point 2 assume that $\mathfrak{A}$ is a $\mathsf{sig}(\varphi_1)$-structure that satisfies φ_1. If $P^{\mathfrak{A}} = \emptyset$, then the desired model $\mathfrak{B}$ is obtained from $\mathfrak{A}$ by interpreting all predicates in $\mathsf{sig}(\varphi_2) \setminus \mathsf{sig}(\varphi_1)$ as empty. Otherwise, take some $a \in P^{\mathfrak{A}}$. We can follow the existential quantifiers in φ_1 to identify a homomorphic image of an infinite tree in $\mathfrak{A}$ with root a whose edges follow the expected pattern and that is labeled in the expected way by the counter C. Since $\mathfrak{A}$ is a model of φ_1, an initial piece of the identified tree represents an accepting computation tree of M on w provided that the predicates S^p_α behave as expected, that is, if there is no defect of the form described above. Since M does not accept w, there must thus be such a defect, that is a path of length 2^n that links a cell of a configuration with the corresponding cell of a subsequent configuration such that the former is labeled with S_α, but the latter is not labeled with S^p_α. All the elements of the path (with the possible exception of the start point and the end point) are labeled with a different value of the counter C and must thus be distinct. Consequently, we can interpret the counter C' and the other symbols in φ_2 to extend $\mathfrak{A}$ to a model of φ_2, as desired. ❑

It follows directly from Lemma 29 that Σ-entailment, Σ-inseparability, and conservative extensions in GF^2, are 2ExpTime-hard.

E 2ATAs and Their Emptiness Problem

The aim of this section is to show that the emptiness problem for 2ATAs can be solved in time exponential in the number of states. For proving this, we reduce it to the emptiness problem of the standard two-way alternating tree automata over trees of fixed outdegree [39].

We start with making precise the semantics of 2ATAs. Let $\mathcal{A} = (Q, \Theta, q_0, \delta, \Omega)$ be a 2ATA and (T, L) a Θ-labeled tree. A *run for $\mathcal{A}$ on (T, L)* is a $T \times Q$-labeled tree (T_r, r) such that:

- $\varepsilon \in T_r$ and $r(\varepsilon) = (\varepsilon, q_0)$;
- For all $y \in T_r$ with $r(y) = (x, q)$ and $\delta(q, L(x)) = \varphi$, there is an assignment v of truth values to the transitions in φ such that v satisfies φ and:
  - if $v(p) = 1$, then $r(y') = (x, p)$ for some successor y' of y in T_r;
  - if $v(\langle - \rangle p) = 1$, then $x \neq \varepsilon$ and there is a successor y' of y in T_r with $r(y') = (x \cdot -1, p)$;
  - if $v([-]p) = 1$, then $x = \varepsilon$ or there is a successor y' of y in T_r such that $r(y') = (x \cdot -1, p)$;
  - if $v(\Diamond p) = 1$, then there is a successor x' of x in T and a successor y' of y in T_r such that $r(y') = (x', p)$;
  - if $v(\Box p) = 1$, then for every successor x' of x in T, there is a successor y' of y in T_r such that $r(y') = (x', p)$.

Let $\gamma = i_0 i_1 \cdots$ be an infinite path in T_r and denote, for all $j \geq 0$, with q_j the state such that $r(i_0 \cdots i_j) = (x, q_j)$. The path γ is *accepting* if the largest number m such that $\Omega(q_j) = m$ for infinitely many j is even. A run (T_r, r) is accepting, if all infinite paths in T_r are accepting. Finally, a tree is accepted if there is some accepting run for it.

We next introduce strategy trees similar to [39, Section 4]. A *strategy tree for a 2ATA $\mathcal{A}$* is a tree (T, τ) where τ labels every node in T with a subset $\tau(x) \subseteq 2^{Q \times (\mathbb{N} \cup \{-1\}) \times Q}$, that is, with a graph with nodes from Q and edges labeled with natural numbers or -1. Intuitively, $(q, i, p) \in \tau(x)$ expresses that, if we reached node x in state q, then we should send a copy of the automaton in state p to $x \cdot i$. For each label ζ, we define $\mathsf{state}(\zeta) = \{q \mid (q, i, q') \in \zeta\}$, that is, the set of sources in the graph ζ. A strategy tree is *on an input tree* (T', L) if $T = T'$, $q_0 \in \mathsf{state}(\tau(\varepsilon))$, and for every $x \in T$, the following conditions are satisfied:

1. if $(q, i, p) \in \tau(x)$, then $x \cdot i \in T$;
2. if $(q, i, p) \in \tau(x)$, then $p \in \mathsf{state}(\tau(x \cdot i))$;
3. if $q \in \mathsf{state}(\tau(x))$, then the truth assignment $v_{q,x}$ defined below satisfies $\delta(q, L(x))$:
   - **a.** $v_{q,x}(p) = 1$ iff $(q, 0, p) \in \tau(x)$;
   - **b.** $v_{q,x}(\langle - \rangle p) = 1$ iff $(q, -1, p) \in \tau(x)$;
   - **c.** $v_{q,x}([-]p) = 1$ iff $x = \varepsilon$ or $(q, -1, p) \in \tau(x)$;
   - **d.** $v_{q,x}(\Diamond p) = 1$ iff there is some i with $(q, i, p) \in \tau(x)$;
   - **e.** $v_{q,x}(\Box p) = 1$ iff $(q, i, p) \in \tau(x)$, for all $x \cdot i \in T$;

A *path* β in a strategy tree (T, τ) is a sequence $\beta = (u_1, q_1)(u_2, q_2) \cdots$ of pairs from $T \times Q$ such that for all $\ell > 0$, there is some i such that $(q_\ell, i, q_{\ell+1}) \in \tau(u_\ell)$ and $u_{\ell+1} = u_\ell \cdot i$. Thus, β is obtained by following moves prescribed by the strategy tree. We say that β is *accepting* if the largest number m such that $\Omega(q_i) = m$, for infinitely many i, is even. A strategy tree (T, τ) is *accepting* if all infinite paths in (T, τ) are accepting.

▶ **Lemma 30.** *A 2ATA accepts a Θ-labeled tree (T, L) iff there is an accepting strategy tree for $\mathcal{A}$ on (T, L).*

Proof. The "if"-direction is immediate: just read off an accepting run from the accepting strategy tree.

For the "only if"-direction, we observe that acceptance of an input tree can be defined in terms of a parity game between Player 1 (trying to show that the input is accepted) and Player 2 (trying to challenge that). The initial configuration is (ε, q_0) and Player 1 begins. Consider a configuration (x, q). Player 1 chooses a satisfying truth assignment v of $\delta(q, L(x))$. Player 2 chooses a transition α with $v(\alpha) = 1$ and the next configuration is determined as follows:

- if $\alpha = p$, then the next configuration is (x, p),
- if $\alpha = \langle - \rangle p$, then the next configuration is $(x \cdot -1, p)$ unless $x = \varepsilon$ in which case Player 1 loses immediately.
- if $\alpha = [-]p$, then the next configuration is $(x \cdot -1, p)$ unless $x = \varepsilon$ in which case Player 2 loses immediately;
- if $\alpha = \Diamond p$, then Player 1 chooses some i with $x \cdot i \in T$ (and loses if no such i exists) and the next configuration is $(x \cdot i, p)$;
- if $\alpha = \Box p$, then Player 2 chooses some i with $x \cdot i \in T$ (and loses if no such i exists) and the next configuration is $(x \cdot i, p)$.

Player 1 wins an infinite play $(x_0, q_0)(x_1, q_1) \cdots$ if the largest number m such that $\Omega(q_i) = m$, for infinitely many i, is even. It is not difficult to see that Player 1 has a winning strategy on an input tree iff $\mathcal{A}$ accepts the input tree. Observe now that the defined game is a parity game and thus Player 1 has a winning strategy iff she is has a *memoryless* winning strategy [15]. It remains to observe that a memoryless winning strategy is nothing else than an accepting strategy tree. ❑

Based on the previuos lemma, we show that, if $L(\mathcal{A})$ is not empty, then it contains a tree of small outdegree.

▶ **Lemma 31.** *If $L(\mathcal{A}) \neq \emptyset$, then there is a $(T, L) \in L(\mathcal{A})$ such that the outdegree of T is bounded by the number of states in $\mathcal{A}$.*

Proof. Let (T, L) be a Θ-labeled tree and τ an accepting strategy tree on T. We construct a tree $T' \subseteq T$ and an accepting strategy tree τ' on (T', L') where L' is the restriction of L to T'. Start with $T' = \{\varepsilon\}$ and τ' the empty mapping. Then exhaustively repeat the following step. Select an $x \in T'$ with $\tau'(x)$ undefined, in a fair way. Then construct $\tau'(x)$ as follows:

1. for every $(q, i, p) \in \tau(x)$ with $i \in \{-1, 0\}$, include (q, i, p) in $\tau'(x)$;
2. for every $p \in Q$, choose an i such that $(q, i, p) \in \tau(x)$ for some q, if existant. Then add $x \cdot i$ to T' and include (q', i, p) in $\tau'(x)$ for all $(q', j, p) \in \tau(x)$;
3. further include (q, i, p) in $\tau'(x)$ whenever $x \cdot i \in T'$ and $(q, j, p) \in \tau(x)$ for all j with $x \cdot j \in T$.

Clearly, T' has the desired outdegree. It remains to show that τ' is an accepting strategy tree on (T', L'). Observe that the following properties hold for all $x \in T'$, and $p, q \in Q$:

(i) $(q, i, p) \in \tau(x)$ iff $(q, i, p) \in \tau'(x)$, for $i \in \{-1, 0\}$;

(ii) $(q, i, p) \in \tau(x)$ for some $i > 0$ with $x \cdot i \in T$ iff $(q, j, p) \in \tau'(x)$ for some $j > 0$ with $x \cdot j \in T$.

Observe that we have $q_0 \in \mathsf{state}(\tau'(\varepsilon))$, by Points (i) and (ii) and since $q_0 \in \mathsf{state}(\tau(\varepsilon))$. It can be verified that Conditions 1 and 2 of a strategy tree being on an input tree are satisfied due to the construction of T' and τ'. For Condition 3, take any $x \in T'$ and $q \in \mathsf{state}(\tau'(x))$. As $q \in \mathsf{state}(\tau(x))$, we know that the truth assignment $v_{q,x}$ defined for τ satisfies $\delta(q, V(x))$. Let $v'_{q,x}$ be the truth assignment for τ', q, x. It suffices to show that, for all transitions α, $v_{q,x}(\alpha) = 1$ implies $v'_{q,x}(\alpha) = 1$. By Point (i), this is the case for transitions of the form $p, \langle - \rangle p, [-]p$. For $\alpha = \Diamond p$, we know that there is some i, p with $(q, i, p) \in \tau(x)$. By Point (ii), we know that $(q, i', p) \in \tau'(x)$ for some i' with $x \cdot i' \in T'$, and thus, $v'_{q,x}(\alpha) = 1$. For $\alpha = \Box p$, we know that $(q, i, p) \in \tau(x)$ for all i with $x \cdot i \in T$. By construction if τ', it follows that $(q, i, p) \in \tau(x)$ for all i with $x \cdot i \in T'$, as required.

We finally argue that τ' is also accepting. Let $\beta = (u_1, q_1)(u_2, q_2) \cdots$ be an infinite path in (T', τ'). We construct an infinite path $\beta' = (u'_1, q_1)(u'_2, q_2)(u'_3, q_3) \cdots$ in (T, τ) as follows:

- $u'_1 = u_1$;
- if $u_{i+1} = u_i \cdot \ell$ with $\ell \in \{-1, 0\}$, then $u'_{i+1} = u'_i \cdot \ell$.
- if $u_{i+1} = u_i \cdot \ell$ for some $\ell \geq 0$ with $(q_i, \ell, q_{i+1}) \in \tau'(x)$, then, by Point (ii), there is some ℓ' with $(q_i, \ell', q_{i+1}) \in \tau(x)$ and $x \cdot \ell' \in T'$. Set $u_{i+1} = u_i \cdot \ell'$.

Since every infinite path in (T, τ) is accepting, so is β', and thus β. ❏

We are now reduce to reduce the emptiness problem of 2ATAs to the emptiness of alternating automata running on trees of fixed outdegree [39], which we recall here. A tree T is *k-ary* if every node has exactly k successors. A *two-way alternating tree automaton over k-ary trees (2ATA^k)* that are Θ-labeled is a tuple $\mathcal{A} = (Q, \Theta, q_0, \delta, \Omega)$ where Q is a finite set of *states*, Θ is the *input alphabet*, $q_0 \in Q$ is an *initial state*, δ is the *transition function*, and $\Omega : Q \to \mathbb{N}$ is a *priority function*. The transition function maps a state q and some input letter θ to a *transition condition* $\delta(q, \theta)$, which is a positive Boolean formula over the truth constants true, false, and transitions of the form $(i, q) \in [k] \times Q$ where $[k] = \{-1, 0, \ldots, k\}$. A *run* of $\mathcal{A}$ on a Θ-labeled tree (T, L) is a $T \times Q$-labeled tree (T_r, r) such that

1. $r(\varepsilon) = (\varepsilon, q_0)$;
2. for all $x \in T_r, r(x) = (w, q)$, and $\delta(q, \tau(w)) = \varphi$, there is a (possibly empty) set $\mathcal{S} = \{(m_1, q_1), \ldots, (m_n, q_n)\} \subseteq [k] \times Q$ such that $\mathcal{S}$ satisfies φ and for $1 \leq i \leq n$, we have $x \cdot i \in T_r$, $w \cdot m_i$ is defined, and $\tau_r(x \cdot i) = (w \cdot m_i, q_i)$.

Accepting runs and accepted trees are defined as for 2ATAs. The emptiness problem for 2ATA^{k}s can be solved in time exponential in the number of states [39].

▶ **Theorem 32.** *The emptiness problem for 2ATAs can be solved in time exponential in the number of states.*

Proof. Let $\mathcal{A} = (Q, \Theta, q_0, \delta, \Omega)$ be a 2ATA with n states. We transform $\mathcal{A}$ into a 2ATA^n $\mathcal{A}' = (Q', \Theta', q'_0, \delta', \Omega)$, running over n-ary Θ'-labeled trees, where $Q' = Q \uplus \{q'_0, q', q_r\}$ and $\Theta' = \Theta \times \{0, 1\}$. The extended alphabet and the extra states q'_0, q', q_r are used to simulate transitions of the form $[-]p$. We make sure that the additional component labels the root node with 1 and all other nodes with 0, and based on this use q_r to check whether we are at the root of the input tree.

Formally, we proceed as follows. For all $q \in Q$, $\theta \in \Theta$, and $b \in \{0, 1\}$ obtain $\delta'(q, (\theta, b))$ from $\delta(q, \theta)$ by replacing q with $(0, q)$, $\langle - \rangle q$ with $(-1, q)$, $[-]q$ with $(0, q_r) \vee (-1, q)$, $\Diamond q$ with $\bigvee_{i=1}^{n}(i, q)$, and $\Box q$ with $\bigwedge_{i=1}^{n}(i, q)$. To enforce the intended labeling in the second component and the correct behaviour for q_r, we set:

$$\delta'(q'_0, (\theta, b)) = \begin{cases} \mathsf{false} & \text{if } b = 0 \\ (0, q_0) \wedge \bigwedge_{i=1}^{k}(i, q') & \text{otherwise} \end{cases}$$

$$\delta'(q', (\theta, b)) = \begin{cases} \bigwedge_{i=1}^{k}(i, q') & \text{if } b = 0 \\ \mathsf{false} & \text{otherwise} \end{cases}$$

$$\delta'(q_r, (\theta, b)) = \begin{cases} \mathsf{true} & \text{if } b = 1 \\ \mathsf{false} & \text{otherwise} \end{cases}$$

Using Lemma 31 it is straightforward to verify that $L(\mathcal{A}) \neq \emptyset$ iff $L(\mathcal{A}') \neq \emptyset$. Since the translation can be done in polynomial time and the emptiness problem for 2ATA^{k}s is in ExpTime, also emptiness for 2ATAs is in ExpTime. ❏